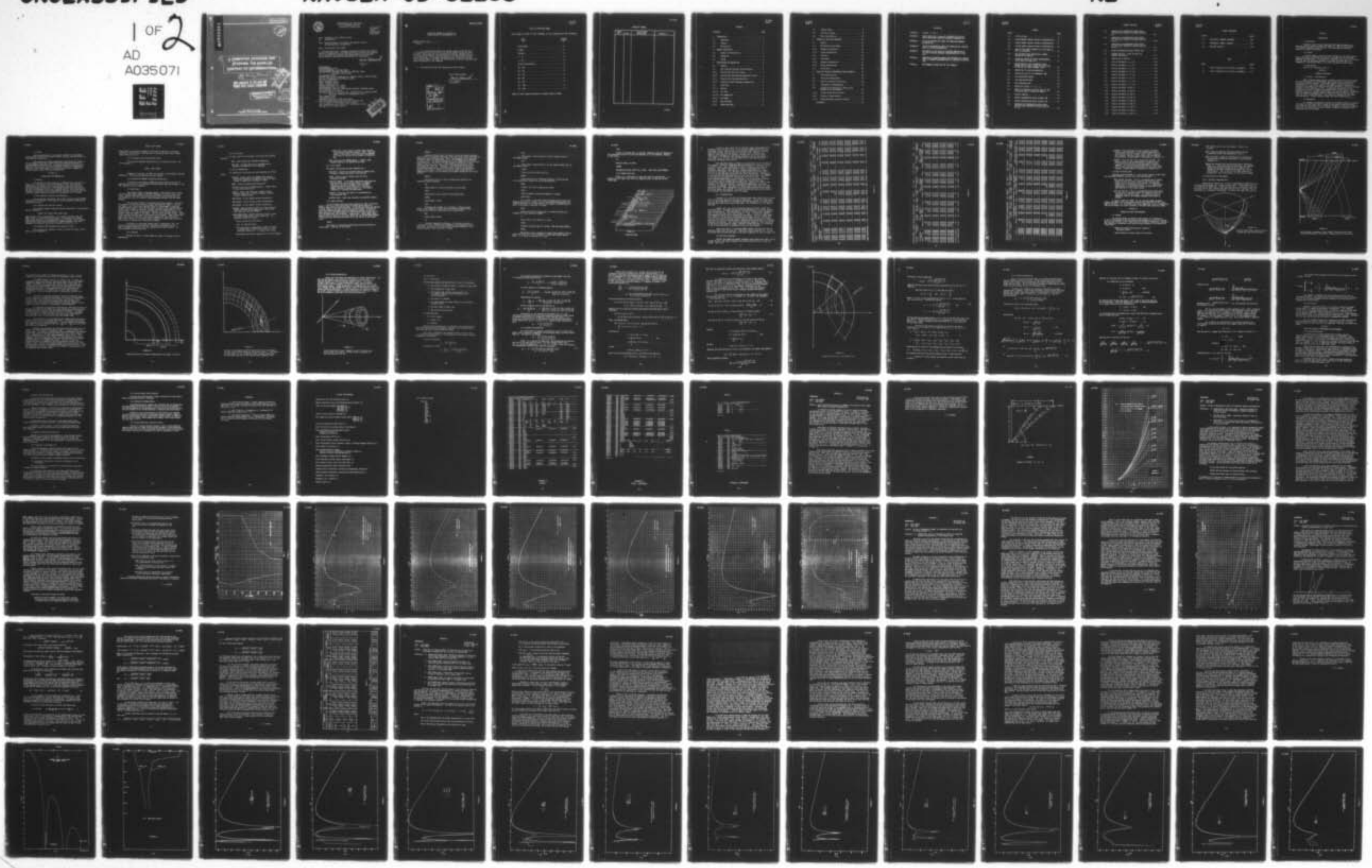
AD-A035 071 NAVAL SEA SYSTEMS COMMAND WASHINGTON D C F/G 17/1
A COMPUTER PROGRAM FOR STUDYING THE DOPPLER CONTENT OF REVERBER--ETC(U)
1976 P MARSH
UNCLASSIFIED NAVSEA-OD-52258 NL

1 OF 2
AD
A035071



(14) NAVSEA-OD-52258

(20) B.S.

(6) A COMPUTER PROGRAM FOR STUDYING THE DOPPLER CONTENT OF REVERBERATION,

(10) Philip/Marsh

(11) 1976

(12) 182

Published by Direction
Commander, Naval Sea System Command

391 345

DEPARTMENT OF THE NAVY
NAVAL UNDERSEA CENTER
SAN DIEGO, CALIFORNIA 92132

IN REPLY REFER TO:
4514/HGP:lb
735086
Ser 451-1118
2 Dec 1976

From: Commander, Naval Undersea Center
To: Distribution

Subj: Computer Program for Studying the Doppler Content of Reverberation; forwarding of

Encl: (1) One copy of OD 52258

1. Ordnance Data 52258, A Computer Program for Studying the Doppler Content of Reverberation, was developed using Torpedo Mk 46 characteristics and approved by Engineering Change Proposal 46-2/NUC/2122. This document is forwarded to distribution for information.

M. O. Heinrich

M. O. HEINRICH
By direction

Distribution:
COMOPTEVFORLANT w/encl
DEPCOMOPTEVFORPAC, San Diego w/encl
NAVSEASYSCOMHQ (SEA-662D, SEA-662D-21, SEA-03B) w/encl
(SEA-0342) w/3 cys encl
ONR (ONR-466) w/encl
PROJMGR ASWS (PM-4), Washington D.C. (ASW-13, ASW-14, ASW-54) w/encl
National Oceanographic Data Center w/encl
NELC, San Diego w/encl
NSRDC, Annapolis w/encl
NSRDC, Carderock w/encl
OCEANAV, Washington D.C. w/encl
NAVSURFWPNCEN, White Oak w/encl
NAVPOSTGRADSCOL (Library, Tech Reports Section), Monterey w/encl
NRL, Washington D.C. w/encl
NAVTORPSTA (Quality Evaluation Lab., Technical Lab.), Keyport w/encl
(Director, Research and Engineering) w/encl
NUSC, Newport w/encl
NUSC NLONLAB w/encl
NAVWPNCEN, China Lake (Code 753) w/encl
Defense Documentation Center (TISIA-1) w/20 cys encl
APL, University of Washington, Seattle w/encl
ARL, Pennsylvania State University, Pennsylvania w/encl
Honeywell Inc., Seattle w/encl

A COMPUTER PROGRAM FOR STUDYING THE DOPPLER CONTENT OF REVERBERATION (U)

NAVORD Ordnance Data __________
A Computer . . . . . .

1. This publication documents the computer program called DOP which is a mathematical tool used in the study of the doppler content of reverberation. The program has been used primarily with the characteristics of the Torpedo MK 46, however, the characteristics of other acoustic systems can be substituted. The program was authored by Mr. Philip Marsh of the Naval Undersea Center, San Diego, and any questions relating to interpretations should be addressed to his attention.

2. This publication does not supersede any other document.

Naval Undersea Center

M. O. Heinrich

M. O. Heinrich
By Direction.

LISTS OF EFFECTIVE PAGES

Total number of pages in this document is 111, consisting of the following:

Sheet No.	#Change No.
Title Sheet	0
A page	0
Flyleaf	0
i - iv	0
1 - 32	0
Initial Distribution	0
A1 - A3	0
B1 - B4	0
C1 - C11	0
D1 - D4	0
E1 - E5	0
F1 - F39	0
G1 - G70	0

#Zero in this column indicates an original page or sheet.

RECORD OF CHANGES

CHANGE NO.	DATE	TITLE OR BRIEF DESCRIPTION	ENTERED BY

CONTENTS

APPENDICES

FIGURES

FIGURES CONTINUES

FIGURES CONTINUES

TABLE

SECTION I

INTRODUCTION

1.1 Background

Torpedoes are the prime conventional antisubmarine weapons and they use acoustic systems to detect the submarine. Some of these systems use frequency dependent characteristics of the returned echo from the submarine to enhance performance.

1.2 Purpose

A digital computer program has been developed at the Naval Undersea Center (NUC) which facilitates the analysis of systems that use frequency dependent characteristics. The program is called DOP and is used in conjunction with two other NUC developed programs called SONAR and RAYSRT which are documented in references, ~~1 and~~ 2. This report documents the DOP program.

1.3 Publications

See list of references, page 32.

SECTION II

GENERAL DESCRIPTION

2.1 Doppler in Reverberation

When a single-frequency pulse is emitted from an active sonar system on a moving platform, the reverberation seen by the system is spread due in large part to doppler effects. Program DOP computes the spectrum of such reverberation in a refractive medium. The doppler content is computed as a function of the speed of the sonar platform and (optionally) of circular turning or motion of the scatterers, or both. In addition, the spectrum of the original pulse can be included as a spreading effect, since even a "single-frequency" pulse has a harmonic content due to its finite duration. The energy level is computed in frequency bands of specified width at specified times relative to the transmitted pulse. Four values are computed for each band/time combination: surface, bottom, volume, and total reverberation.

2.2 Boundaries

For boundary reverberation (surface or horizontal bottom), increments are summed in random phase from all areas of the boundary returning energy in a given band at a given time. Scattering strength is a function of grazing angle, and all combinations of paths (direct, refracted, reflected) to the scattering areas are included.

2.3 Volume

Volume reverberation is, at present, computed in an unbounded uniform medium, summing contributions from all volumes returning energy in a given band at a given time.

Both boundary and volume computations consider two-way losses in the environment and average transmit and receive beam-pattern losses to each incremental scattering unit. In addition, optional filtering can be applied to each band and TVG (Time Varied Gain) action can be applied at each time. Also, total energy in all bands at each time is computed for surface, bottom, volume, and total reverberation.

SECTION III

INSTRUCTIONS FOR RUNNING DOP

3.1 General

The program is written primarily in FORTRAN IV for execution on a UNIVAC 1110. The program is not self-contained in that there are some functions and subroutines that must be supplied by the user to match the particular system being studied. (The program has been exercised here at NUC with the characteristics of the Torpedo MK 46 Mod 1 and an experimental torpedo.) A description of these user supplied routines, the input data, and ancillary programs necessary or useful to the execution of DOP follows.

3.2 User Supplied Functions and Subroutines

For any specific application, one or more of the following FORTRAN functions will be required. Since they are vehicle dependent, they must be supplied by the user.

3.2.1 Function OXL (Off-axis Losses)

This routine computes transducer pattern attenuation in any direction. The call sequence is:

```
VALUE = OXL (IFLAG, COSA, COSB, COSC)
```

where IFLAG is 0 for receive pattern, 1 for transmit and COSA, COSB, and COSC are the X, Y, and Z direction cosines in the direction of interest. Other values necessary for the computation, e.g. transducer type, frequency, sound velocity, etc. may be supplied via a COMMON statement. The value returned is a fraction of the on-axis intensity, from 0.0 to 1.0.

3.2.2 Function RRF (Reverberation Rejection Filter)

This routine will interpose a filter to modify the energy in each band. The call sequence is:

VALUE = RRF (FREQ)

where FREQ is the center frequency of the band in question. The value returned is the fraction of energy which is passed by the filter in that band, from 0.0 to 1.0.

3.2.3 Function TVG (Time Variable Gain)

This routine computes receiver gain as a function of time. The call sequence is:

GAIN = TVGF (TIME)

Examples of OXL, RRF, and TVGF are provided in the program listings, Appendix G. These are unrelated to any real system.

3.2.4 Subroutine SPRCMP (Spreading Computation)

In addition to the above, SPRCMP may be provided by the user to generate one or more of the spreading function tables when the program is run. The subroutine has no arguments.

3.3 Input Data

Input data to DOP is from two sources: the output file of sorted ray data from programs SONAR and RAYSRT (references 1 & 2) and input data cards. Besides the sorted ray data returned from insonified portions of the boundaries, the file contains eleven parameters passed from the SONAR program.

The card input format is free-form with blanks ignored. Variables are punched in fields of arbitrary length, separated by commas. Data may be integer, real (including a decimal point) or alpha-numeric (appearing between single quotes). Note that no check is made for the appropriateness of any piece of data to any name. Neither names nor numbers may be split between cards. All data cards appear literally in the printed output. Two kinds of data may appear on the cards: option fields and data fields. Option fields contain only the name of the option to be invoked. These options may modify the form or content of input data, computation, and output data. Data fields consist of a variable name followed by an equal sign and one or more values, as appropriate, separated by commas.

For option or data names longer than six characters, only the first six characters are interpreted. Additional characters may be used to improve readability, but are ignored by the input routine. All data variables are initially zero.

3.3.1 Options

Options for DOP are listed below for input, for output, and for computation.

3.3.1.1 For Input

For input, there are two options, which have the following meanings:

- GO - Stop reading data and begin computation.
- NO TAPE - No input data file is provided and no boundary reverberation will be computed.

3.3.1.2 For Computation

For computation, the options and their meanings are listed below:

- CENTER - Doppler bands are computed such that the transmitted frequency is centered in one of the bands instead of appearing at the edge of a band.
- END - Stop all program activity and exit.
- FILTER (Used only with SPREAD option) - Apply filter to the spread output data.
- KNOTS - Compute the intensity in bands of equal apparent range rate or "knots of doppler" instead of equal frequency range.
- NO BOTTOM - Do not compute bottom reverberation.
- NO SURFACE - Do not compute surface reverberation.
- NO VOLUME - Do not compute volume reverberation.
- SPREAD - Apply spreading function to output data and change the format of the output data listing. (See note under data variable BSPRED.)
- TIME COMPUTATION - Compute additional values of time for which reverberation is to be determined. The additional values of time are:
  - Ping interval (See PING below.)
  - 1/2 Δt (See DELT below.)
  - For every path or combination of paths to surface or bottom whose earliest arrival time is t, the additional values are: t, $t+1/2\Delta t$, and $t-1/2\Delta t$.
  - Seventeen fixed values ranging from .01 to 2.0 seconds.

- Additional values every 1/2 second from 2 seconds to the value of the PING interval, NOTE: The total number of time entries read from cards and computed as a result of the use of this option is limited to 400.

- TVG (used only with SPREAD option) - Apply a time-varying function to the spread output data.

3.3.1.3 For Output

For output, the options and their meanings are:

- NO PRINT - Suppress the printed output of doppler data. Printing of input data cards is not affected.

- PLOT - Write a tape of doppler data for use by subsequent programs.

- RELATIVE BANDS - If the KNOTS computation option has not been specified, print the band limits in kilohertz relative to the source frequency; with the KNOTS computation option, print band limits in knots of doppler relative to vehicle speed; i.e. zero doppler (unspread) is returned from dead ahead.

- TOTALS - Print only the totals of reverberation at each requested time.

- (PRINT EVERY - Under Data Variables also modifies output.)

3.3.2 Data Variables

In the following listing of data variables for DOP, an asterisk (which is NOT part of the name) denotes variables whose values are normally taken from the input data tape. If the same variables are supplied on cards, however, the data from the cards would be used. It should be noted that some values which could be changed are implicit in the ray data supplied by the SONAR program and that changing these values by card would be meaningless and misleading. Examples are: source depth (DO), sound velocity of medium at the source depth (CO) and bottom depth (DBTTM). (Please note that the second character of CO, DO, and FO is the numeral zero.)

ALPHC*

The product of attenuation coefficient and sound velocity at source depth. Units are dB per second.

BSPRED

Spreading function table for bottom. Since the bottom scatterers would usually be considered stationary, this table would usually represent the spectrum of the transmitted pulse. NOTE: The appearance in the data deck of BSPRED, SSPRED, or VSPRED makes the use of the SPREAD option redundant and unnecessary. Each table is entered as half a symmetrical table of an odd number of entries. First value is the proportion of energy in a band remaining after spreading. Next value is the proportion spread to the two neighboring bands, etc. Therefore, twice the total of all values entered should equal one plus the first value. Each of the tables has a maximum of 150 values. Failure to include all three tables with the SPREAD option causes a call to SPRCMP in an attempt to generate the missing table(s).

BWIDTH

Band width in hertz or knots as appropriate, based on the computation option selected.

CO*

Sound velocity in yards per second at source depth.

DATE*

Alpha-numeric date, maximum of two machine words.

DBTTM*

Bottom depth in feet.

DELT

Effective pulse length, Δt, in seconds. Effective pulse length is the length of a square pulse with the same energy content as the pulse of interest, which may not be a square pulse.

DO*

Source depth in feet.

FO

Transmit frequency in kilohertz. If no value is supplied, 1 kilohertz is used. Although this parameter is not supplied from the input tape, it is implicit in the ray data from the attenuation values (and the spreading loss correction, if used).

IDC*

Alpha-numeric identification as on the "constant card" of the SONAR program.

IDV*

Alpha-numeric identification as on the "semi-variable card" of the SONAR program.

LOGMV*

Volume scattering coefficient, dB.

NBEAM

Coded description of transducer patterns, if more than one can be generated; intended for use by subroutine OXL.

OMEGA

Platform turn rate in degrees per second.

PING*

Interval between successive transmits in seconds.

PRINT EVERY

For a value n, print only every nth doppler band at each time. Bands to be printed are chosen so as to include the band containing the transmit frequency as its lower bound or its center, if CENTER option is used. If no value is supplied, 1 is used.

PULSE

Coded description of pulse shape, if several options exist. Intended for use by subroutine SPRCMP.

S*

Source level in dB relative to 1 yard.

SSPRED

Spreading function table for surface. (See note under BSPRED.)

THTMAX

Approximate value in degrees of largest angle (between velocity vector and sound rays) to be considered in computing unspread doppler bands. If no value is supplied, 90 is used.

TIME

Values of elapsed time, in seconds, (measured from the midpoint of the transmitted pulse) at which reverberation is to be computed. Maximum of 400 values.

VS

Vehicle speed, in knots.

VSPRED

Spreading function table for volume. (See note under BSPRED).

3.3.3 Sample Data Deck

Figure 3-1 illustrates an input card deck for program DOP. It assumes that at least four files (or complete sets) of data are on the input tape.

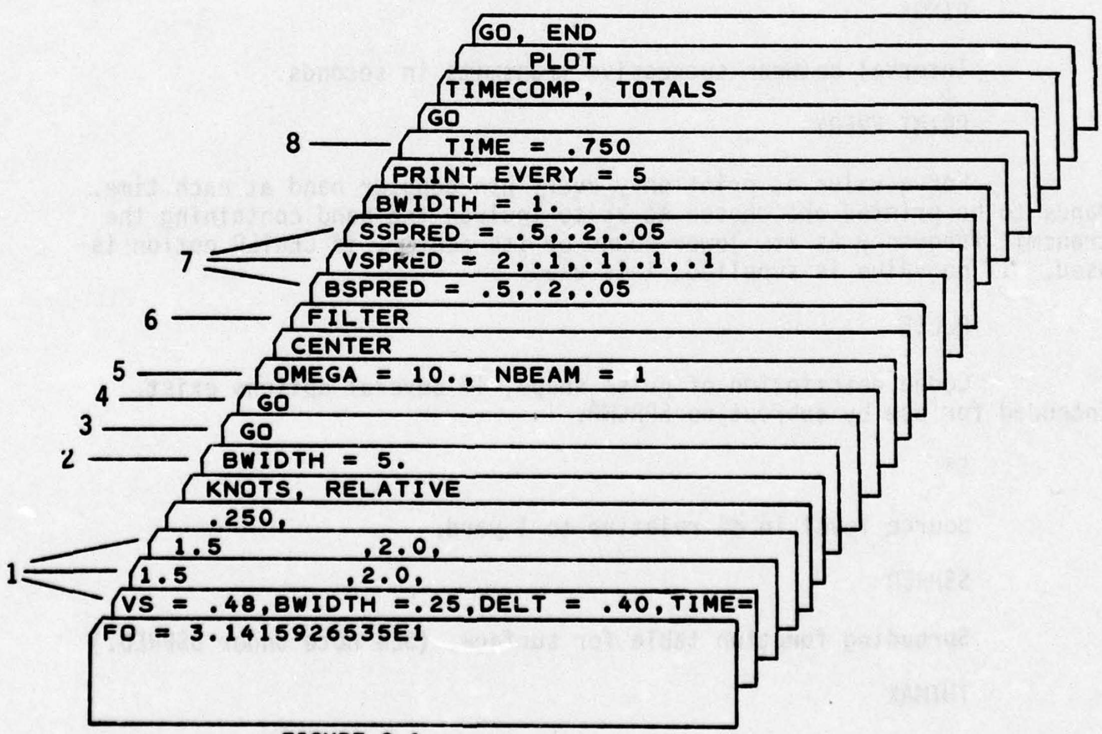


FIGURE 3-1

A DOP DATA DECK

In such a case, each file on the input tape (corresponding to different "semi-variable" cards in the original SONAR program deck) is processed in turn. Options and data variables from cards are retained until over written from the tape or from new cards. (Please note, however, that options once invoked or turned "on" can not be turned "off" again.) In general, the program proceeds until an END option is read from a card, or until the last data file on the tape has been processed, whichever occurs first.

The indicated cards in Figure 3-1 illustrate certain features of the program. Cards at 1 exemplify the continuation of a string of values on several cards. Also, note that of the six values of time, two are repeated. The program sorts values of time in ascending order and eliminates duplicates. The card at 2 redefines band width as 5. knots although it had been set to .25 knots on the second card of the deck. Card at 3 marks the end of card data for file 1 on the tape. The card at 4 indicates that all the same data is to be used for File 2. At 5 NBEAM is used because the hypothetical torpedo has two transmit beams. Some sonar problems might never use this feature. At 6 invocation of FILTER will cause a call to require RRF to modify the level of each band. Without this option, no RRF would be required. Similarly, if one of the variables at 7 was not supplied, a subroutine SPRCMP would have been called by the program. The card at 8 replaces all of the values of TIME previously read. Note throughout, the free, even arbitrary use of blanks which, we repeat, are ignored. Also note that values used on these cards are arbitrary and unrelated to any real system.

3.4 Program Output

Figures 3-2, 3-3, and 3-4 illustrate the output from Files 1, 3, and 4 of an input tape used with the sample input deck. In addition, the input deck, up to the GO option, or between successive GO options, is listed on a separate page before each output data page. No illustration of this page is provided.

Figures 3-2, 3-3, and 3-4 show the three principal formats for the output data. The pages are mostly self-explanatory. One point, however, should be mentioned. The appearance of .00 dB in a column seems ambiguous, since it may mean zero dB or zero energy (- dB). In practice, however, this ambiguity should rarely present any difficulty. The truly empty bands will always lie at the top or the bottom of a column of data. Unless the first or last nonvoid band has exactly zero dB, which can usually be determined from the band and column totals, leading and trailing zeros in a column represent zero energy. This form of printout was chosen in lieu of printing an arbitrary large negative number, because it made the page less cluttered. It has to date presented no problems to users of the program.

Three data decks, to programs SONAR, RAYSRT, and DOP were used to generate Figures 3-2, 3-3 and 3-4. These are listed in Appendix A. It will be noticed that the DOP data deck is not quite identical to that in Figure 3-1.

3.5 Ancillary Programs

Besides the SONAR and RAYSRT programs which produce the input file of sorted ray data, two other programs exist which may be of help to the user of DOP. These are:

PROGRAM 8Q0003 -- DOPPLER CONTENT OF REVERBERATION IDC REVERB IDV DOPLER DATE 07/23/76 PAGE 1 OF 1

V.S, KTS.	C.O, YDS./SEC.	F.O, KHZ.	D.O, FT.	S, DB.	XI, DEG.	OMEGA, DEG./SEC.	P.I., SEC.	DEL. T, SEC.	D. BTM., FT.
65.00	1625.41	31.6159	750.0000	100.00	-2.000	.000	2.0000	.0400	4000.0000

FREQUENCY BAND, KILOHERTZ		TIME = .25000000				TIME = 1.50000000				TIME = 2.00000000			
FROM-	TO-	SURFACE	BOTTOM	VOLUME	TOTAL	SURFACE	BOTTOM	VOLUME	TOTAL	SURFACE	BOTTOM	VOLUME	TOTAL
32.4109	32.4059	.00	.00	-23.28	-23.28	.00	.00	-51.15	-51.15	.00	.00	-58.56	-58.56
32.4059	32.4009	.00	.00	-26.63	-26.63	.00	.00	-54.50	-54.50	.00	.00	-61.91	-61.91
32.4009	32.3959	.00	.00	-29.63	-29.63	.00	.00	-57.49	-57.49	.00	.00	-64.91	-64.91
32.3959	32.3909	.00	.00	-32.47	-32.47	.00	.00	-60.33	-60.33	.00	.00	-67.75	-67.75
32.3909	32.3859	.00	.00	-35.26	-35.26	.00	.00	-63.12	-63.12	.00	.00	-70.56	-70.56
32.3859	32.3809	.00	.00	-38.06	-38.06	.00	.00	-65.92	-65.92	.00	.00	-73.34	-73.34
32.3809	32.3759	.00	.00	-40.91	-40.91	.00	.00	-68.78	-68.78	.00	.00	-76.19	-76.19
32.3759	32.3709	.00	.00	-43.87	-43.87	.00	.00	-71.73	-71.73	.00	.00	-79.15	-79.15
32.3709	32.3659	.00	.00	-46.95	-46.95	-72.03	.00	-74.82	-70.19	.00	.00	-82.23	-82.23
32.3659	32.3609	.00	.00	-50.21	-50.21	-71.34	.00	-78.08	-70.50	.00	.00	-85.49	-85.49
32.3609	32.3559	.00	.00	-53.71	-53.71	-77.60	.00	-81.57	-76.14	.00	.00	-88.99	-88.99
32.3559	32.3509	.00	.00	-57.50	-57.50	-82.58	.00	-85.36	-80.74	.00	.00	-92.78	-92.78
32.3509	32.3459	.00	.00	-61.70	-61.70	-87.61	.00	-89.56	-85.47	.00	.00	-96.98	-96.98
32.3459	32.3409	.00	.00	-66.47	-66.47	-93.17	.00	-94.34	-90.71	.00	.00	-101.75	-101.75
32.3409	32.3359	.00	.00	-72.05	-72.05	-99.87	.00	-99.92	-96.88	.00	.00	-107.33	-107.33
32.3359	32.3309	.00	.00	-77.12	-77.12	-108.99	.00	-104.99	-103.53	.00	.00	-112.40	-112.40
32.3309	32.3259	.00	.00	-75.85	-75.85	-122.84	.00	-103.72	-103.66	.00	.00	-111.13	-111.13
32.3259	32.3209	.00	.00	-72.35	-72.35	-141.65	.00	-100.22	-100.22	.00	.00	-107.63	-107.63
32.3209	32.3159	.00	.00	-69.54	-69.54	-120.97	.00	-97.41	-97.39	.00	.00	-104.82	-104.82
32.3159	32.3109	.00	.00	-67.31	-67.31	-111.57	.00	-95.18	-95.08	.00	.00	-102.59	-102.59
32.3109	32.3059	.00	.00	-65.49	-65.49	-106.04	.00	-93.35	-93.13	.00	.00	-100.77	-100.77
32.3059	32.3009	.00	.00	-63.98	-63.98	-102.53	.00	-91.85	-91.49	.00	.00	-99.26	-99.26
32.3009	32.2959	.00	.00	-62.74	-62.74	-99.95	.00	-90.60	-90.12	.00	.00	-98.02	-98.02
32.2959	32.2909	.00	.00	-61.70	-61.70	-98.01	.00	-89.57	-88.99	.00	.00	-96.98	-96.98
32.2909	32.2859	.00	.00	-60.86	-60.86	-96.55	.00	-88.72	-88.06	.00	.00	-96.14	-96.14
32.2859	32.2809	.00	.00	-60.17	-60.17	-95.41	.00	-88.03	-87.30	.00	.00	-95.45	-95.45
32.2809	32.2759	.00	.00	-59.61	-59.61	-94.53	.00	-87.48	-86.69	.00	.00	-94.89	-94.89
32.2759	32.2709	.00	.00	-59.18	-59.18	-93.85	.00	-87.04	-86.22	.00	.00	-94.46	-94.46
TOTAL REVERBERATION		.00	.00	-20.39	-20.39	-67.88	.00	-48.25	-48.21	.00	.00	-55.67	-55.67

Figure 3-2

PROGRAM 800003 -- DOPPLER CONTENT OF REVERBERATION IDC REVERB IDV DOPL 2 DATE 07/23/76

V.S, KTS.	C.O, YDS./SEC.	F.O, KHZ.	D.O, FT.	S, DB.	XI, DEG.	OMEGA, DEG./SEC.	P.I., SEC.	DEL. T, SEC.	D. BTM., FT.
45.00	1636.70	31.4159	500.0000	100.00	-2.000	10.000	2.0000	.0400	4000.0000

DOPPLER BAND, KNOTS		PURE TONE, TIME = .75000000				AFTER SPREADING, TIME = .75000000				SPREAD WITH FILTER, TIME = .75000000			
FROM-	TO-	SURFACE	BOTTOM	VOLUME	TOTAL	SURFACE	BOTTOM	VOLUME	TOTAL	SURFACE	BOTTOM	VOLUME	TOTAL
5.5000	4.5000	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.5000	-.5000	.00	.00	.00	.00	-61.60	.00	-48.62	-48.41	-101.60	.00	-88.62	-88.41
-4.5000	-5.5000	-57.17	.00	-58.75	-54.88	-55.55	.00	-48.43	-47.66	-95.55	.00	-88.43	-87.66
-9.5000	-10.5000	-65.89	.00	-66.94	-63.37	-64.33	.00	-64.09	-61.20	-104.33	.00	-104.09	-101.20
-14.5000	-15.5000	-78.55	.00	-77.50	-74.98	-76.63	.00	-75.29	-72.90	-116.63	.00	-115.29	-112.90
-19.5000	-20.5000	-77.96	.00	-76.78	-74.32	-78.17	.00	-77.32	-74.72	-118.17	.00	-117.32	-114.72
-24.5000	-25.5000	-84.81	.00	-84.25	-81.51	-84.10	.00	-82.40	-80.15	-124.10	.00	-122.40	-120.15
-29.5000	-30.5000	-92.36	.00	-90.03	-88.03	-91.32	.00	-88.41	-86.62	-131.32	.00	-128.41	-126.62
-34.5000	-35.5000	-89.16	.00	-86.00	-84.29	-89.33	.00	-86.47	-84.66	-129.33	.00	-126.47	-124.66
-39.5000	-40.5000	-89.64	.00	-86.61	-84.86	-89.75	.00	-86.80	-85.02	-129.75	.00	-126.80	-125.02
-44.5000	-45.5000	-95.58	.00	-89.87	-88.83	-94.34	.00	-89.63	-88.37	-134.34	.00	-129.63	-128.37
-49.5000	-50.5000	-101.06	.00	-95.29	-94.27	-100.71	.00	-94.49	-93.56	-119.36	.00	-113.14	-112.21
-54.5000	-55.5000	-108.88	.00	-103.38	-102.30	-108.06	.00	-101.57	-100.69	-123.73	.00	-117.23	-116.35
-59.5000	-60.5000	-120.76	.00	-116.18	-114.88	-118.63	.00	-111.71	-110.91	-132.56	.00	-125.65	-124.84
-64.5000	-65.5000	-126.05	.00	-121.79	-120.40	-125.89	.00	-121.65	-120.26	-138.61	.00	-134.36	-132.98
-69.5000	-70.5000	-128.01	.00	-123.87	-122.45	-128.07	.00	-123.93	-122.51	-139.84	.00	-135.71	-134.29
-74.5000	-75.5000	-137.85	.00	-135.01	-133.19	-136.21	.00	-131.04	-129.89	-147.22	.00	-142.05	-140.90
-79.5000	-80.5000	-142.21	.00	-138.86	-137.21	-142.28	.00	-139.74	-137.82	-152.65	.00	-150.11	-148.19
-84.5000	-85.5000	-145.60	.00	-146.01	-142.79	-145.36	.00	-143.19	-141.13	-155.18	.00	-153.01	-150.95
-89.5000	-90.5000	.00	.00	-150.08	-150.08	-161.67	.00	-152.72	-152.20	-171.01	.00	-162.05	-161.53
-94.5000	-95.5000	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
TOTAL	REVERBERATION	-45.36	.00	-38.46	-37.65	-45.36	.00	-38.46	-37.65	-85.35	.00	-78.46	-77.64

FIGURE 3-3

OD 52258

PROGRAM 800003 — DOPPLER CONTENT OF REVERBERATION IDC REVERB IDV DOPL 3 DATE 07/23/76 PAGE 1 OF 1

V.S, KTS.	C.O, YDS./SEC.	F.O, KHZ.	D.O, FT.	S, DB.	XI, DEG.	OMEGA, DEG./SEC.	P.I., SEC.	DEL. T, SEC.	D. BTM., FT.
65.00	1679.30	31.4159	275.0000	100.00	-2.000	10.000	2.0000	.0400	4000.0000

TOTAL REVERBERATION FROM 101 BANDS, EACH OF 1.000 KNOTS BANDWIDTH

	PURE TONE				AFTER SPREADING				SPREAD WITH FILTER AND TVG			
TIME	SURFACE	BOTTOM	VOLUME	TOTAL	SURFACE	BOTTOM	VOLUME	TOTAL	SURFACE	BOTTOM	VOLUME	TOTAL
.02000000	.00	.00	1.13	1.13	.00	.00	1.13	1.13	.00	.00	-48.48	-48.48
.05000000	.00	.00	-8.34	-8.34	.00	.00	-8.34	-8.34	.00	.00	-57.44	-57.44
.08918396	.00	.00	-13.89	-13.89	.00	.00	-13.89	-13.89	.00	.00	-62.41	-62.41
.10000000	-15.98	.00	-14.99	-12.45	-15.98	.00	-14.99	-12.45	-53.74	.00	-63.36	-53.29
.10918396	-15.90	.00	-15.83	-12.85	-15.90	.00	-15.83	-12.85	-53.59	.00	-64.08	-53.22
.12918395	-15.80	.00	-17.45	-13.56	-15.80	.00	-17.45	-13.54	-53.32	.00	-65.45	-53.06
.20000000	-26.71	.00	-21.74	-20.56	-26.71	.00	-21.74	-20.54	-73.82	.00	-68.94	-67.72
.30000000	-31.57	.00	-25.90	-24.86	-31.57	.00	-25.90	-24.86	-77.82	.00	-72.18	-71.13
.40000000	-25.75	.00	-29.00	-24.07	-25.75	.00	-29.00	-24.07	-71.28	.00	-74.52	-69.59
.50000000	-24.96	.00	-31.53	-24.08	-24.96	.00	-31.53	-24.08	-69.82	.00	-76.40	-68.96
.60000000	-25.87	.00	-33.69	-25.20	-25.87	.00	-33.69	-25.20	-70.19	.00	-77.99	-69.52
.70000000	-27.37	.00	-35.59	-26.76	-27.37	.00	-35.59	-26.76	-71.19	.00	-79.40	-70.58
.75000000	-28.58	.00	-36.47	-27.92	-28.58	.00	-36.47	-27.92	-72.17	.00	-80.05	-71.51
.80000000	-29.42	.00	-37.31	-28.77	-29.42	.00	-37.31	-28.77	-72.80	.00	-80.67	-72.14
.90000000	-32.61	.00	-38.88	-31.69	-32.61	.00	-38.88	-31.69	-75.57	.00	-81.83	-74.65
1.00000000	-35.24	.00	-40.33	-34.07	-35.24	.00	-40.33	-34.07	-77.84	.00	-82.92	-76.66
1.20000000	-39.67	.00	-42.98	-38.01	-39.67	.00	-42.98	-38.01	-81.61	.00	-84.91	-79.94
1.40000001	-44.11	.00	-45.39	-41.69	-44.11	.00	-45.39	-41.69	-85.47	.00	-86.75	-83.05
1.50918451	-45.20	.00	-46.63	-42.85	-45.20	.00	-46.63	-42.85	-86.29	.00	-87.71	-83.93
1.52918451	-45.50	-68.71	-46.86	-43.10	-45.50	-68.71	-46.86	-43.10	-86.53	-98.67	-87.88	-83.99
1.54918450	-46.21	-65.83	-47.08	-43.59	-46.21	-65.83	-47.08	-43.59	-87.20	-96.52	-88.06	-84.32
1.59999999	-48.27	-67.81	-47.65	-44.91	-48.27	-67.81	-47.65	-44.91	-89.13	-101.25	-88.50	-85.67
1.61836843	-48.82	-68.88	-47.85	-45.28	-48.82	-68.88	-47.85	-45.28	-89.64	-102.51	-88.66	-86.01
1.63836843	-49.37	-70.16	-48.08	-45.65	-49.37	-70.16	-48.08	-45.65	-90.14	-103.70	-88.84	-86.35
1.65836842	-49.86	-71.44	-48.30	-45.99	-49.86	-71.44	-48.30	-45.99	-90.58	-104.75	-89.01	-86.65
1.72755235	-51.28	-73.47	-49.09	-47.03	-51.28	-73.47	-49.09	-47.03	-91.85	-107.38	-89.65	-87.55
1.74755235	-51.64	-72.28	-49.32	-47.30	-51.64	-72.28	-49.32	-47.30	-92.16	-106.16	-89.83	-87.77
1.76755235	-51.95	-71.42	-49.55	-47.56	-51.95	-71.42	-49.55	-47.56	-92.43	-105.37	-90.02	-87.97
1.80000000	-52.43	-71.13	-49.93	-47.97	-52.43	-71.13	-49.93	-47.97	-92.84	-106.34	-90.33	-88.33
2.00000000	-54.96	-70.84	-52.26	-50.35	-54.96	-70.84	-52.26	-50.35	-94.96	-109.21	-92.25	-90.33

FIGURE 3-4

- DENFSP - This program may be used to prepare spreading function tables for DOP. It will combine (convolve) a spectrum and a density function and/or convert a spectrum into a density function. Print, punch, and plot options are available. Program DENSFP is documented in reference (3).

- SRNBT4 - This uses the 'plot' output tape from DOP. The program prints or plots reverberation in doppler bands. It can translate the data into broader bands than were computed. It can also plot selected bands vs. time, and plot or punch cards for enabling level for use by the SONAR program. Program SRNBT4 is documented in reference (3).

3.6 Other Considerations

Although standard FORTRAN IV is the primary language of DOP, there are two UNIVAC FORTRAN V features which have been employed:

- The INCLUDE Statement - This is a convenient way of adding blocks of cards to several routines, insuring that such blocks will be identical in each routine. For compilers without this or an analogous feature, it is a comparatively simple matter to duplicate the blocks of cards and to add them physically in the proper places.

- The FLD Function - This is a bit-manipulating function which is used only in subroutine INPUT for assembling individual characters into variable names. For compilers without a comparable feature, an assembly-language routine must be written.

As used in subroutine INPUT, the two constants NWORD and NCHAR contain the number of bits per machine word and the number of bits per alphanumeric character. For the UNIVAC 1110, these numbers are 36 and 6 respectively. They can be changed via a data statement in the BLOCK DATA subroutine, INBLK.

SECTION IV

THEORETICAL BASIS AND PROCEDURE

4.1 General

The thinking which led through various models of reverberation to the one implemented in DOP has been covered extensively in earlier informal works by A.B. Poynter. Five of these works are inlcuded as Appendices B through F. The conventions, assumptions, and simplifications employed in the program are listed below:

- Both the surface and bottom are treated as horizontal planes.

- The platform's velocity vector is horizontal.

- The medium varies only with depth. There is no current.
- All times are measured from the midpoint of the transmitted pulse (see Appendices C and D).
- No allowance is made for differences in transmit and receive paths caused by translation of the platform (Appendix E).
- When the transmit and receive paths are of different types, e.g. one direct and one reflected, the boundary scattering strength between paths is taken as the average of the back scattering strengths for the two single paths. This is one of the least defensible assumptions, and investigation in this area is needed.

4.2 Description of the Models

4.2.1 Boundary Reverberation

Refer now to Figure 4-1. A ray is considered as emanating from a source at some depth, d_o, located on the Z axis. Since sound velocity is a function of depth only, a ray lies entirely in a vertical plane, and the locus of all rays leaving the source at an initial angle θ_o from the horizontal is a surface of revolution. If these rays reach the surface,

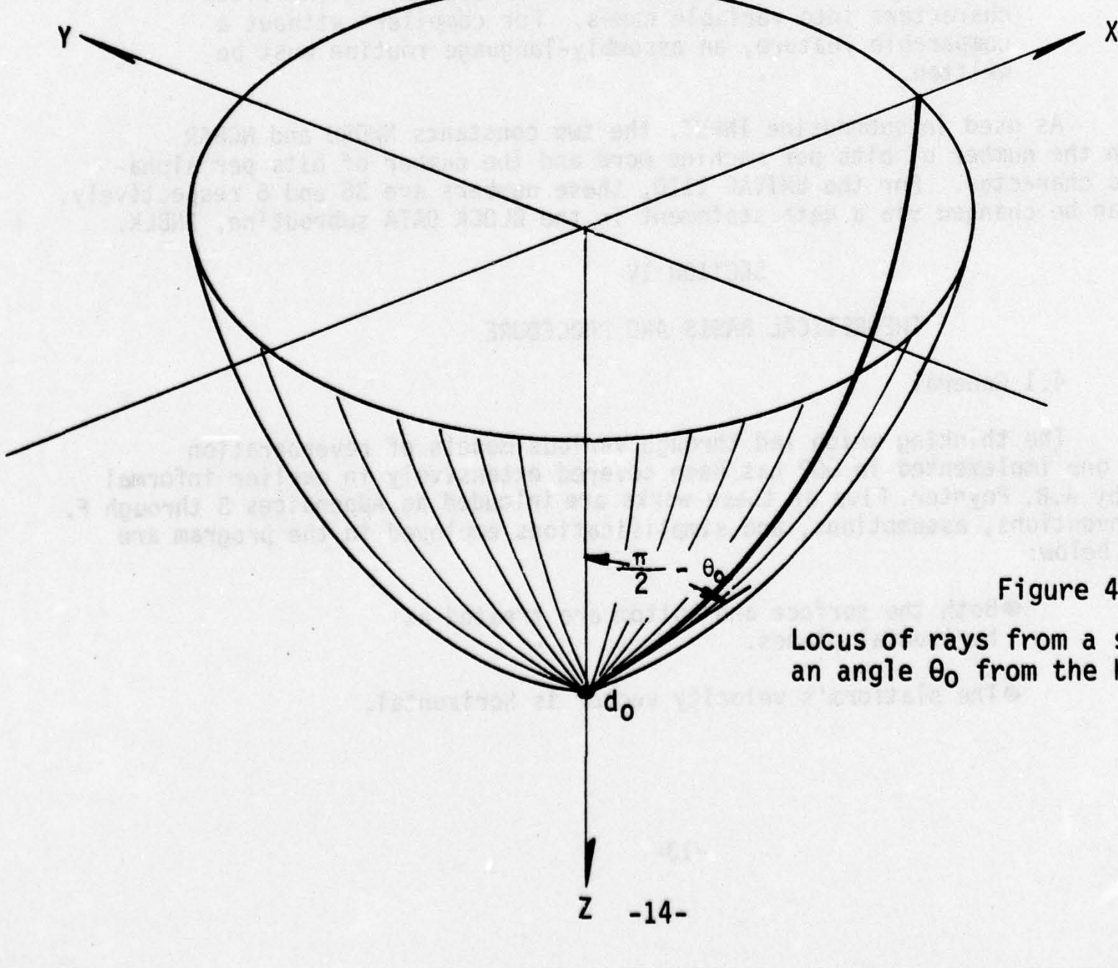


Figure 4-1

Locus of rays from a source at d_o at an angle θ_o from the horizontal.

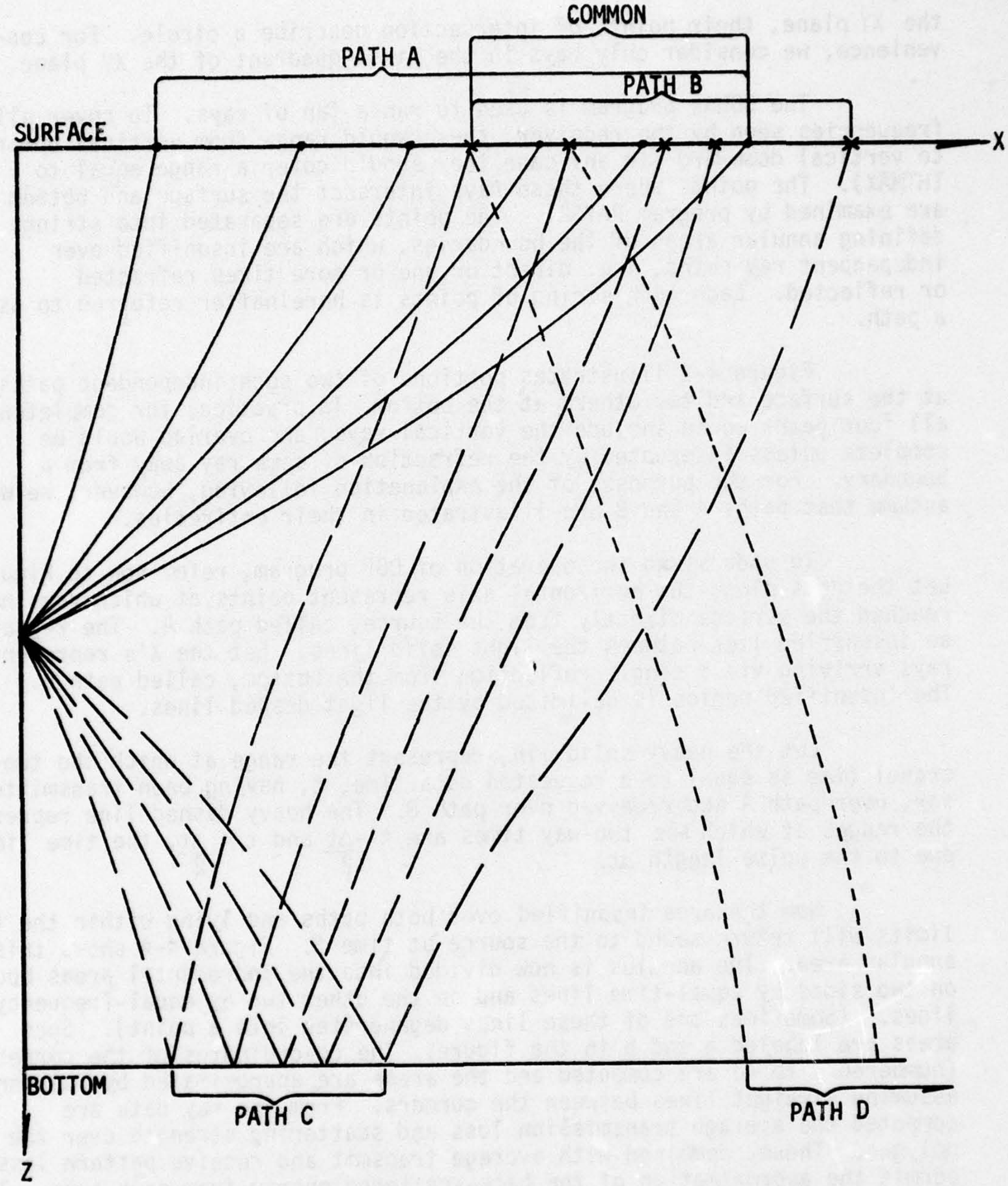


FIGURE 4-2

Two different ray-bundles, called "paths" (direct path and singly bottom-reflected path) insonifying a common portion of the surface.

the XY plane, their points of intersection describe a circle. For convenience, we consider only rays in the first quadrant of the XY plane.

The SONAR program is used to run a fan of rays. To cover all frequencies seen by the receiver, these would range from vertical upward to vertical downward (In any case they should cover a range equal to ± THTMAX). The points where these rays intersect the surface and bottom are examined by program RAYSRT. The points are separated into strings defining annular areas of the boundaries, which are insonified over independent ray paths, i.e. direct or one or more times refracted or reflected. Each such string of points is hereinafter referred to as a path.

Figure 4-2 illustrates portions of two such independent paths at the surface and two others at the bottom. In practice, for completeness, all four paths would include the vertical rays, and overlap would be complete unless interupted by the refraction of some ray away from a boundary. For the purposes of the explanation following, however, we will assume that paths A and B are illustrated in their entireties.

To understand the operation of DOP program, refer now to Figure 4-3. Let the dots along the horizontal axis represent points at which rays have reached the surface directly from the source, called path A. The region so insonified lies between the light solid lines. Let the X's represent rays arriving via a single reflection from the bottom, called path B. The insonified region is delimited by the light dashed lines.

Let the heavy solid line represent the range at which the two-way travel time is equal to a requested data time, t, having been transmitted, say, over path A and received over path B. The heavy dashed line represents the ranges at which the two-way times are $t - \frac{\Delta t}{2}$ and $t + \frac{\Delta t}{2}$, the time limits due to the pulse length Δt.

Now the area insonified over both paths and lying within the time limits will return sound to the source at time t. Figure 4-4 shows this annular area. The annulus is now divided into the incremental areas bounded on two sides by equal-time lines and on the other two by equal-frequency lines. (Sometimes one of these lines degenerates into a point). Such areas are labeled a and b in the figure. The co-ordinates of the corners (numbered 1 to 4) are computed and the areas are approximated by polygons, assuming straight lines between the corners. From the ray data are computed the average transmission loss and scattering strength over the polygon. These, combined with average transmit and receive pattern losses, permit the approximation of the back-scattered energy from each area. These contributions, expressed in intensities, are summed in random phase within the requested time and band limits.

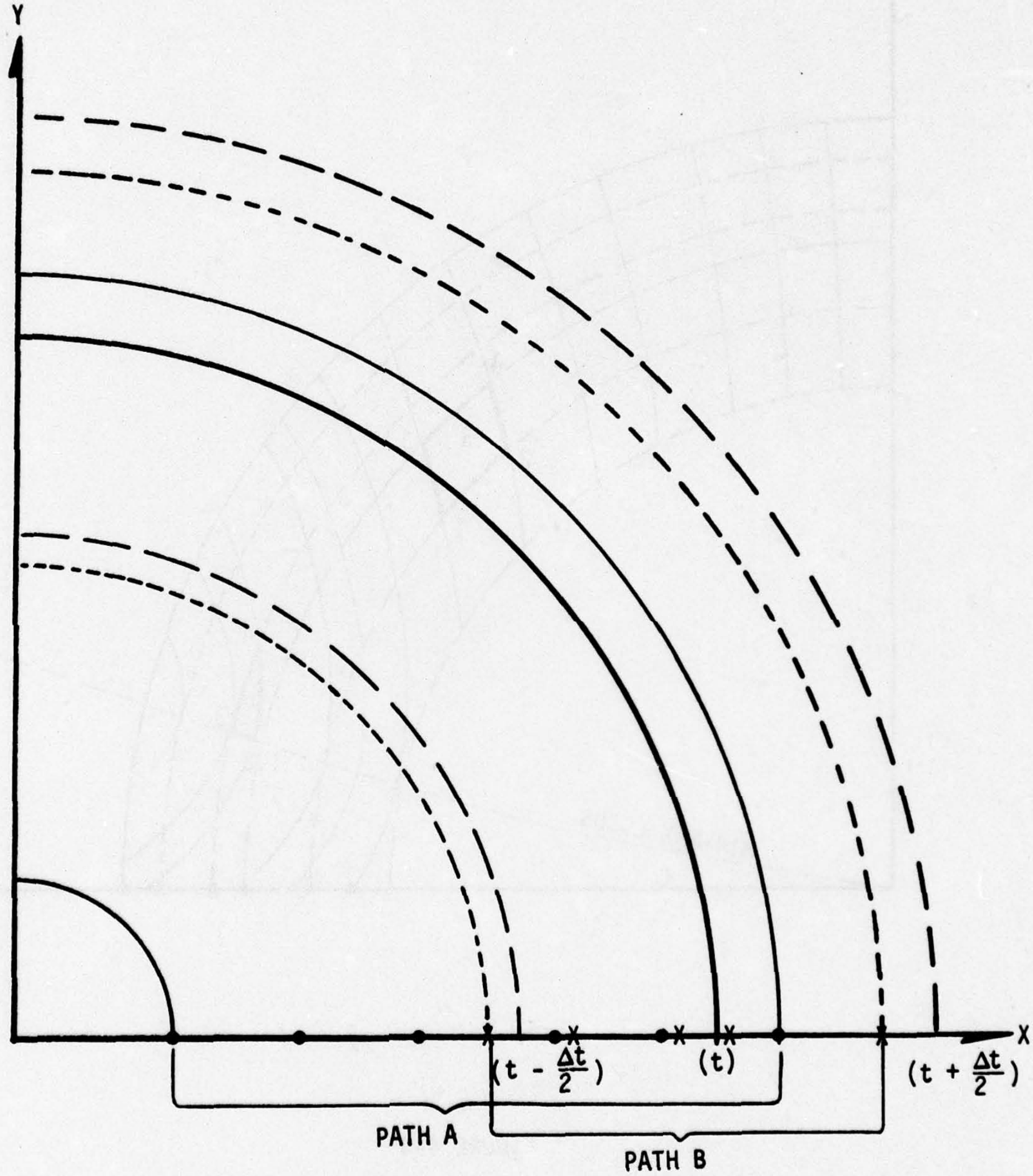


FIGURE 4-3

Insonified portion of surface corresponding to ray paths in Figure 6.

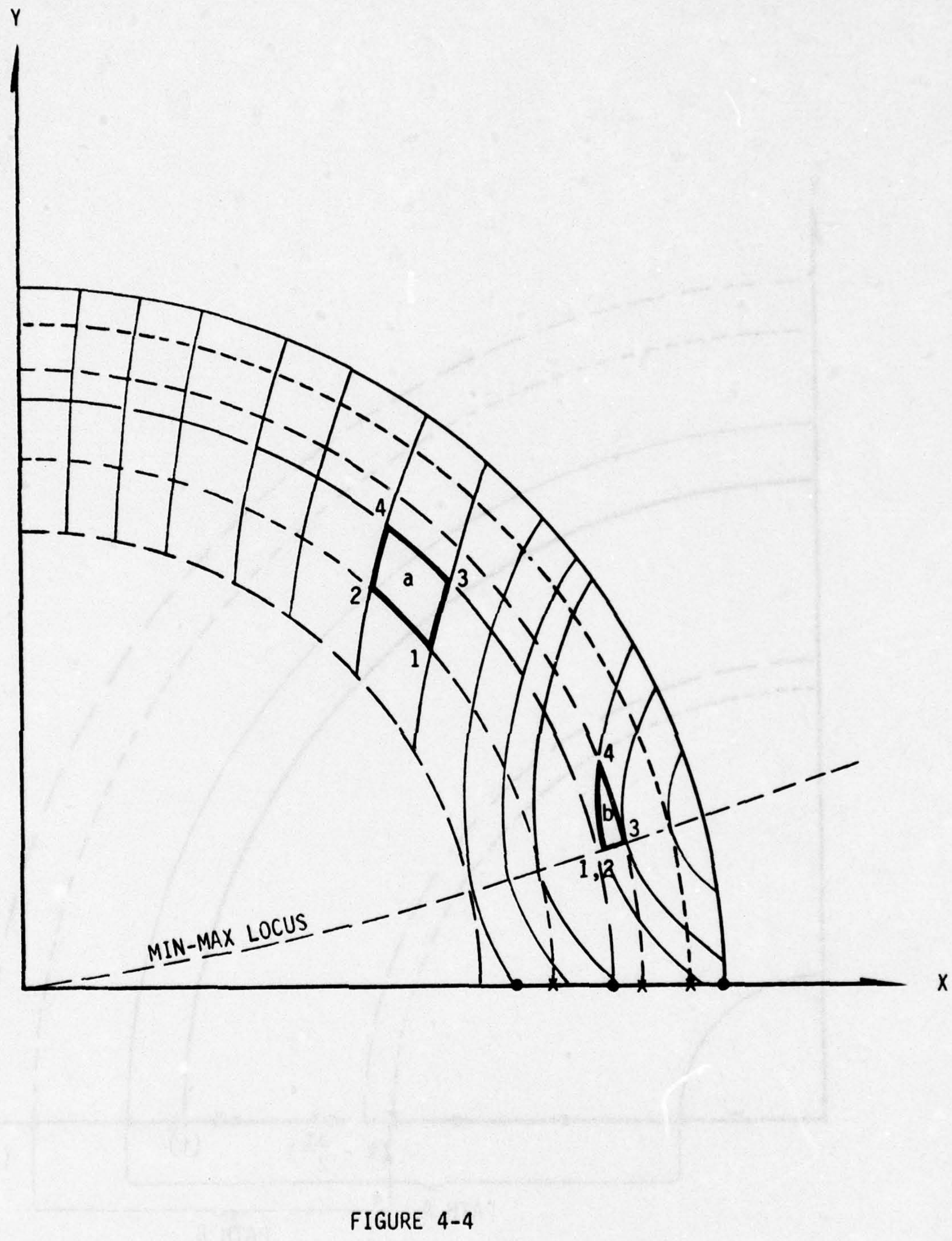


FIGURE 4-4

The part of the common insonified region which also lies between the time limits returns energy to the source at time t. This region has been broken up into incremental areas, bounded on two sides by equal-time lines and on two sides by equal-frequency lines.

4.2.2 Volume Reverberation

Figure 4-5 illustrates the computation of volume reverberation. For the straight-running case, the return in a band at a given time, t, is from the frustum of a hollow cone with spherical bases. The cone is divided radially about its face forming small areas, and the pattern losses to each area are combined to yield an average pattern loss for the iso-frequency volume. Volume reverberation is computed using this pattern loss, average two way transmission losses, volume of the hollow frustum, and volume scattering coefficient. In the turning case the insonified volume is not a cone, and the transmit and receive directions are different. However, the basic procedure is the same as for the non-turning case.

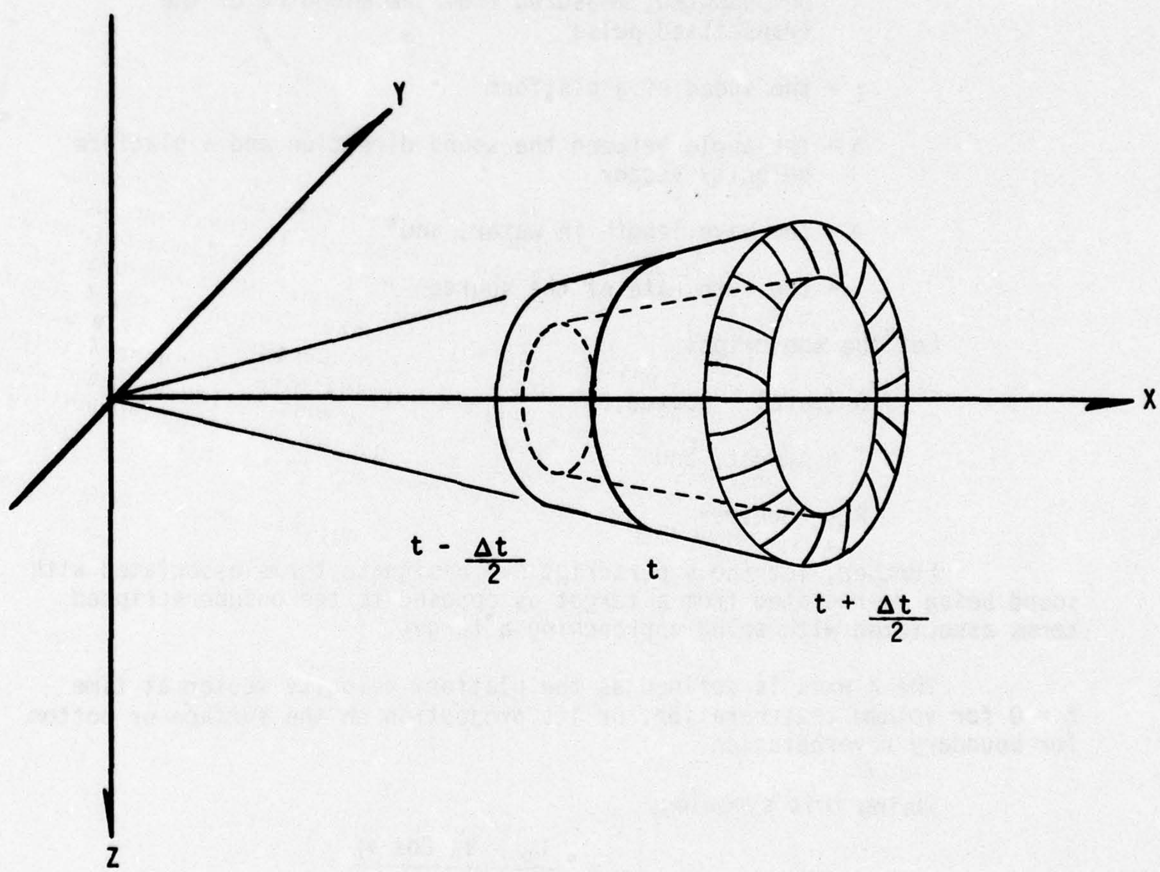


FIGURE 4-5

Volume insonified at time t between two equal-frequency cones and two equal-time spheres. Pattern losses are measured to each area marked on the end.

4.3 Equations

4.3.1 Preliminary

The following symbols and definitions are used in this section:

C = the velocity of sound in water, a function of water depth

f = the frequency of a sound wave

t = the elapsed time at which reverberation is to be computed, measured from the midpoint of the transmitted pulse

V = the speed of a platform

γ = the angle between the sound direction and a platform velocity vector

λ = the wave length in water, and

ω = the turn rate of the source

Let the subscripts

0 (zero) = source,

T = target, and

R = receiver

Further, let the superscript "*" designate terms associated with sound being re-radiated from a target as opposed to the unsuperscripted terms associated with sound approaching a target.

The X axis is defined as the platform velocity vector at time t = 0 for volume reverberation, or its projection on the surface or bottom for boundary reverberation.

Using this symbology

$$\lambda_0 = \frac{C_0 - V_0 \cos \gamma_0}{f_0}$$

Approaching a target

$$\lambda_T = \frac{C_T}{C_0} \lambda_0 = \frac{C_T (C_0 - V_0 \cos \gamma_0)}{C_0 f_0}$$

The frequency observed by a receiver on the target, and that re-radiated into the water will be

$$f_T = \frac{C_T - V_T \text{ Cos } \gamma_T}{\lambda_T} = f_0 \frac{C_0 (C_T - V_T \text{ Cos } \gamma_T)}{C_T (C_0 - V_0 \text{ Cos } \gamma_0)}$$

The wave length of re-radiated sound is

$$\lambda_T^* = \frac{C_T^* - V_T \text{ Cos } \gamma_T^*}{f_T} = \frac{C_T (C_0 - V_0 \text{ Cos } \gamma_0) \; (C_T^* - V_T \text{ Cos } \gamma_T^*)}{f_0 C_0 (C_T - V_T \text{ Cos } \gamma_T)}$$

approaching the receiver,

$$\lambda_R = \frac{C_R}{C_T^*} \lambda_T^* = \frac{C_R C_T (C_0 - V_0 \text{ Cos } \gamma_0) \; (C_T^* - V_T \text{ Cos } \gamma_T^*)}{f_0 C_0 C_T^* (C_T - V_T \text{ Cos } \gamma_T)}$$

and the frequency seen by the receiver is

$$f_R = \frac{C_R - V_R \text{ Cos } \gamma_R}{\lambda_R} = f_0 \frac{C_0 C_T^* (C_T - V_T \text{ Cos } \gamma_T) \; (C_R - V_R \text{ Cos } \gamma_R)}{C_R C_T (C_0 - V_0 \text{ Cos } \gamma_0) \; (C_T^* - V_T \text{ Cos } \gamma_T^*)}$$

Simplifying, in the vicinity of the target, $C^*_T = C_T$, and considering reverberation as being from motionless scatterers (their motion is handled statistically as "spreading" in the returned sound) $V_T = 0$. Also, we are interested only in the monostatic case (source and receiver at the same point), so $C_R = C_0$ and $V_R = V_0$. Applying these identities we have

$$f_R = f_0 \frac{C_0 - V_0 \text{ Cos } \gamma_R}{C_0 - V_0 \text{ Cos } \gamma_0} \qquad (1)$$

4.3.2 Boundary Reverberation

For calculations of boundary reverberation, we will resolve the angle γ into its spherical components, the horizontal angle ϕ and the vertical angle, θ. The equation now becomes

$$f_R = \frac{C_0 - V_0 \text{ Cos } \theta_R \text{ Cos } \phi_R}{C_0 - V_0 \text{ Cos } \theta_0 \text{ Cos } \phi_0}$$

Since C is a function of depth only, the transmitted and received rays are in the same vertical plane. For the straight-running case, $\phi_R = \pi + \phi_0$ but in general $\phi_R = \pi + \phi_0 - \omega t$. Finally, then, the frequency of reverberation received from a point on a boundary,

$$f_R = f_0 \frac{C_0 + V_0 \text{ Cos } \theta_R \text{ Cos } (\phi_0 - \omega t)}{C_0 - V_0 \text{ Cos } \theta_0 \text{ Cos } \phi_0} \qquad (2)$$

Refer again to Figure 4-4. In order to be certain that all incremental areas are assigned to the correct frequency bands, it is important to know what iso-frequency line is tangent to an iso-time line. This is equivalent to finding the maximum or minimum frequency occurring at a given time. It is clear from Figure 4-4 that all the variables of equation (2) except ϕ_0 have been given fixed values. Therefore, we find at what azimuth angle received frequency is a maximum or minimum. Differentiating equation (2),

$$\frac{df_R}{d\phi_0} = -f_0 \frac{V_0 \operatorname{Cos} \theta_R \operatorname{Sin} (\phi_0 - \omega t)}{C_0 - V_0 \operatorname{Cos} \theta_0 \operatorname{Cos} \phi_0}$$

$$-f_0 \frac{C_0 + V_0 \operatorname{Cos} \theta_R \operatorname{Cos} (\phi_0 - \omega t)}{(C_0 - V_0 \operatorname{Cos} \theta_0 \operatorname{Cos} \phi_0)^2} V_0 \operatorname{Cos} \theta_0 \operatorname{Sin} \phi_0 = 0$$

Clearing fractions and rearranging terms,

$$V_0^2 \operatorname{Cos} \theta_0 \operatorname{Cos} \theta_R \left[\operatorname{Sin} \phi_0 \operatorname{Cos} (\phi_0 - \omega t) - \operatorname{Cos} \phi_0 \operatorname{Sin} (\phi_0 - \omega t)\right]$$
$$+ C_0 V_0 \left[\operatorname{Cos} \theta_R \operatorname{Sin} (\phi_0 - \omega t) + \operatorname{Cos} \theta_0 \operatorname{Sin} \phi_0\right] = 0$$

Reducing the first term and expanding the second term using the sin (a-b) identity,

$$V_0^2 \operatorname{Cos} \theta_0 \operatorname{Cos} \theta_R \operatorname{Sin} \omega t$$
$$+ C_0 V_0 (\operatorname{Cos} \theta_R \operatorname{Sin} \phi_0 \operatorname{Cos} \omega t - \operatorname{Cos} \theta_R \operatorname{Cos} \phi_0 \operatorname{Sin} \omega t + \operatorname{Cos} \theta_0 \operatorname{Sin} \phi_0) = 0$$

Again rearranging terms

$$(\operatorname{Cos} \theta_R \operatorname{Cos} \omega t + \operatorname{Cos} \theta_0) \operatorname{Sin} \phi_0 - \operatorname{Cos} \theta_R \operatorname{Sin} \omega t \operatorname{Cos} \phi_0$$
$$+ \frac{V_0}{C_0} \operatorname{Cos} \theta_0 \operatorname{Cos} \theta_R \operatorname{Sin} \omega t = 0$$

Substituting

$$a = \operatorname{Cos} \theta_R \operatorname{Cos} \omega t + \operatorname{Cos} \theta_0$$

$$b = \operatorname{Cos} \theta_R \operatorname{Sin} \omega t \quad , \quad \text{and}$$

$$c = \frac{V_0}{C_0} \operatorname{Cos} \theta_0 \operatorname{Cos} \theta \operatorname{Sin} \omega t \quad ,$$

we have

$$a \operatorname{Sin} \phi_0 - b \operatorname{Cos} \phi_0 + c = 0$$

Squaring and substituting for $\sin^2 \phi_0$ we arrive at the quadratic

$$(a^2 + b^2) \operatorname{Cos}^2 \phi_0 - 2bc \operatorname{Cos} \phi + c^2 - a^2 = 0$$

Set into the quadratic formula and simplified, this becomes finally

$$\text{Cos } \phi_0 = \frac{bc \pm a \sqrt{a^2 + b^2 - c^2}}{a^2 + b^2} \tag{4}$$

To understand the significance of the sign of the radical, consider that when $\omega = 0$, $\sin \omega t = 0$, and $\cos \phi_0 = \pm 1$. That is, the locus of the points of tangency between iso-time and iso-frequency lines (the min-max locus) is the X axis. Also, in this case the iso-frequency line for $f = f_0$ is the Y axis, with greater frequencies to the right, lesser frequencies to the left. Thus it is seen that the positive and negative signs define ϕ_0 for frequency maximum and minima, respectively.

We will now find the co-ordinates of the corners of the insonified polygon labeled 1, 2, 4, 3 in Figure 4-6. At time t and frequency f , from equation (2),

$$f_R C_0 - f_R V_0 \text{ Cos } \theta_0 \text{ Cos } \phi_0 = f_0 C_0 + f_0 V_0 \text{ Cos } \theta_R \text{ Cos } (\phi_0 - \omega t) \quad \text{, or}$$

$$f_0 \text{ Cos } \theta_R \text{ Cos } (\phi_0 - \omega t) + f_R \text{ Cos } \theta_0 \text{ Cos } \phi_0 - \frac{f_R C_0 - f_0 C_0}{V_0} = 0 \quad \text{, or}$$

$$f_0 \text{ Cos } \theta_R (\text{Cos } \omega t \text{ Cos } \phi_0 + \text{Sin } \omega t \text{ Sin } \phi_0) + f_R \text{ Cos } \theta_0 \text{ Cos } \phi_0 - \frac{C_0}{V_0}(f_R - f_0) = 0 \quad \text{, or}$$

$$(f_0 \text{ Cos } \theta_R \text{ Cos } \omega t + f_R \text{ Cos } \theta_0) \text{ Cos } \phi_0 + (f_0 \text{ Cos } \theta_R \text{ Sin } \omega t) \text{ Sin } \phi_0 - \frac{C_0}{V_0}(f_R - f_0) = 0$$

Setting

$$a = f_0 \text{ Cos } \theta_R \text{ Cos } \omega t + f_R \text{ Cos } \theta_0 \text{ ,}$$

$$b = f_0 \text{ Cos } \theta_R \text{ Sin } \omega t \quad \text{, and}$$

$$c = \frac{C_0}{V_0}(f_R - f_0) \quad \text{,}$$

we have

$$a \text{ Cos } \phi_0 + b \text{ Sin } \phi_0 - c = 0$$

Squaring and substituting for $\sin^2 \phi_0$ as previously, we obtain the quadratic

$$(a^2 + b^2) \text{ Cos } \phi - 2ac \text{ Cos } \phi_0 + c^2 - b^2 = 0,$$

and by quadratic formula

$$\text{Cos } \phi_0 = \frac{ac \pm b \sqrt{a^2 + b^2 - c^2}}{a^2 + b^2}$$

OD 52258

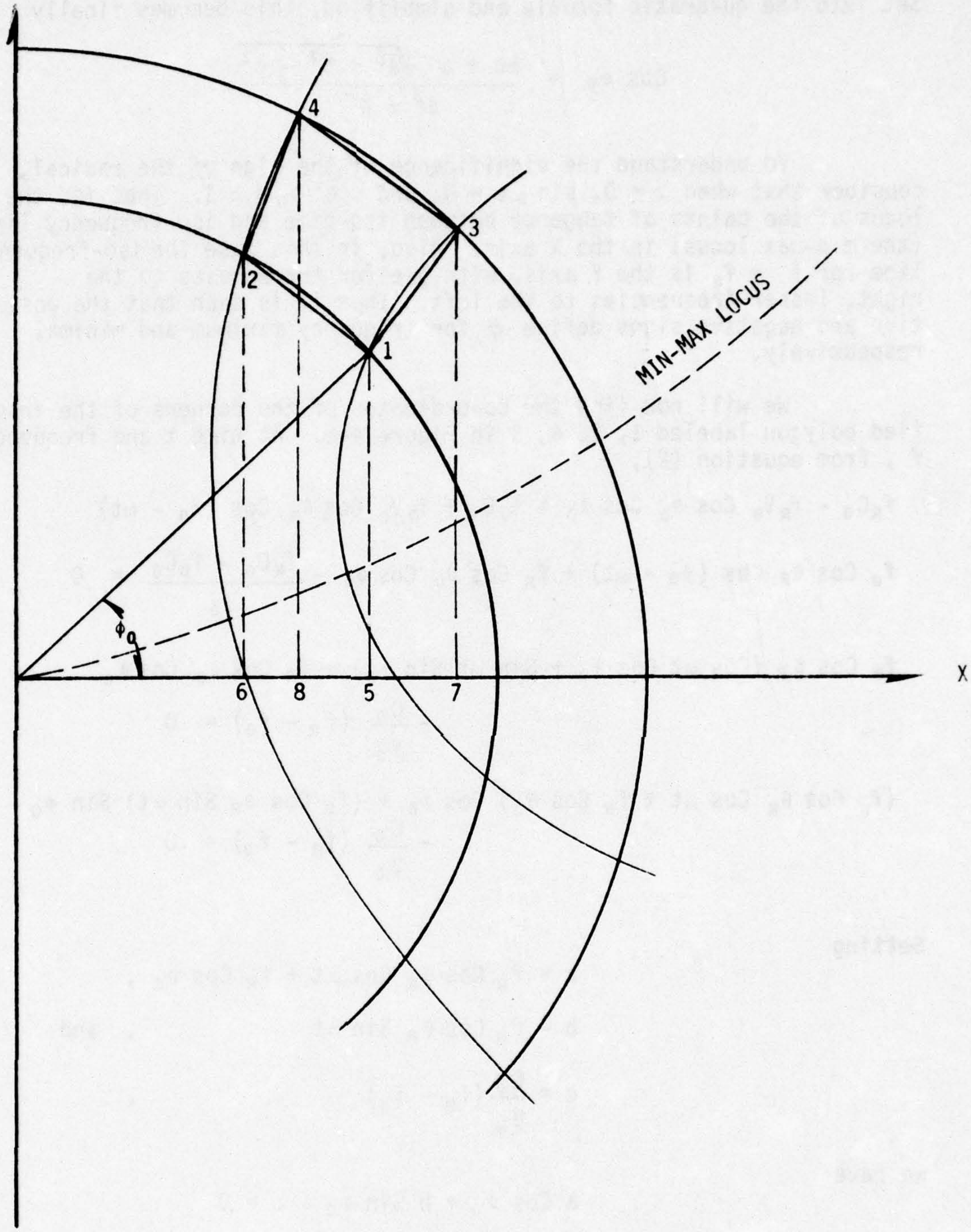


FIGURE 4-6

Finding the value of an incremental area.

Similarly it can be shown that

$$\text{Sin } \phi_0 = \frac{bc \mp a \sqrt{a^2 + b^2 - c^2}}{a^2 + b^2}$$

opposite polarity of the radicals being required by the $\text{Sin } \phi^2 + \text{Cos } \phi^2 = 1$ identity.

Now from Figure 4-6, it is seen that for point 1

$$\text{Cos } \phi_0 = \frac{X_1}{X_t} \quad \text{and} \quad \text{Sin } \phi_0 = \frac{Y_1}{X_t}$$

where x_t is the x value associated with time t, i.e. x_t is the radius of the iso-time circle. Therefore for point 1

$$X_1 = X_t \frac{ac \pm b \sqrt{a^2 + b^2 - c^2}}{a^2 + b^2} \quad \text{and} \tag{5}$$

$$Y_1 = X_t \frac{bc \mp a \sqrt{a^2 + b^2 - c^2}}{a^2 + b^2}$$

From considering the geometry when $\omega = 0$, it can be seen that the upper signs are used for insonified polygons which are clockwise of the min-max locus when $f_R > f_0$, counter clockwise of the locus when $f_R < f_0$, and conversely for the lower signs.

The area of the polygon in Figure 4-6 is clearly the sum of trapezoids 2, 4, 8, 6 and 4, 3, 7, 8 less the sum of 2, 1, 5, 6 and 1, 3, 7, 5. That is,

$$A = \tfrac{1}{2}(X_4 - X_2)(Y_4 + Y_2) + \tfrac{1}{2}(X_3 - X_4)(Y_3 + Y_4)$$
$$- \tfrac{1}{2}(X_1 - X_2)(Y_1 + Y_2) - \tfrac{1}{2}(X_3 - X_1)(Y_3 + Y_1) \quad \text{or}$$

$$A = \tfrac{1}{2}(X_4Y_4 + X_4Y_2 - X_2Y_4 - X_2Y_2 + X_3Y_3 + X_3Y_4 - X_4Y_3 - X_4Y_4$$
$$- X_1Y_1 - X_1Y_2 + X_2Y_1 + X_2Y_2 - X_3Y_3 - X_3Y_1 + X_1Y_3 + X_1Y_1)$$

Consolidating, we arrive at the computational form used in the program:

$$A = \left| \tfrac{1}{2} \left[(X_4Y_2 + X_3Y_4 + X_2Y_1 + X_1Y_3) - (X_2Y_4 + X_4Y_3 + X_1Y_2 + X_3Y_1) \right] \right| \tag{6}$$

The taking of absolute value avoids negative areas in some quadrants.

Equation 6 is quite general and computes correct areas even for triangles.

4.3.3 Volume Reverberation

Now in the straight-running case, as has been mentioned before, the iso-frequency surface is a cone. In the turning case, this surface becomes quite complex, even discontinuous for certain combinations of frequency and turn angle. We will now derive a method of expressing the volume within an iso-frequency surface in the general case.

To begin with, equation (2) is equally applicable to volume reverberation. However, in the simplified model, using a uniform unbounded medium, the transmit and receive paths will necessarily be the same for the straight running case. In any case, θ_R will equal θ_0. Thus the equation becomes

$$f_R = f_0 \frac{C_0 + V_0 \cos\theta_0 \cos(\phi_0 - \omega t)}{C_0 - V_0 \cos\theta_0 \cos\phi_0}$$

clearing the fraction and rearranging terms

$$\cos\theta_0 \left[f_0 \cos(\phi_0 - \omega t) + f_R \cos\phi_0\right] = \frac{C_0}{V_0}(f_R - f_0)$$

Substituting

$$\cos(\phi_0 - \omega t) = \cos\phi_0 \cos\omega t + \sin\phi_0 \sin\omega t,$$

$$\cos\phi_0 = \frac{X}{\sqrt{X^2 + Y^2}},$$

$$\sin\phi_0 = \frac{Y}{\sqrt{X^2 + Y^2}}, \text{ and}$$

$$\cos\theta_0 = \frac{\sqrt{X^2 + Y^2}}{\sqrt{X^2 + Y^2 + Z^2}}, \text{ we obtain}$$

$$\frac{\sqrt{X^2 + Y^2}}{\sqrt{X^2 + Y^2 + Z^2}}\left[f_0\left(\frac{X}{\sqrt{X^2 + Y^2}}\cos\omega t + \frac{Y}{\sqrt{X^2 + Y^2}}\sin\omega t\right) + f_R \frac{X}{\sqrt{X^2 + Y^2}}\right] = \frac{C_0}{V_0}(f_R - f_0)$$

or

$$f_0 (X \cos\omega t + Y \sin\omega t) + f_R X = \frac{C_0}{V_0}(f_R - f_0)\sqrt{X^2 + Y^2 + Z^2}$$

or

$$X(f_0 \cos t + f_R) + Y(f_0 \sin\omega t) = \frac{C_0}{V_0}(f_R - f_0)\sqrt{X^2 + Y^2 + Z^2} \qquad (7)$$

Equation (7) describes the iso-frequency surface in cartesian coordinates.

For simplicity let us substitute

$$a = f_0 \text{ Cos } \omega t + f_R \quad ,$$

$$b = f_0 \text{ Sin } \omega t \quad , \text{ and}$$

$$c = \frac{C_0}{V_0}(f_R - f_0) \quad , \text{ yielding}$$

$$aX + bY = c\sqrt{X^2 + Y^2 + Z^2} \qquad (8)$$

By rotating axes through some angle δ, the Y term of the above equation can be eliminated. *Using the superscript "prime" to denote measurements* in a rotated system and the transformations

$$X' = X \text{ Cos } \delta + Y \text{ Sin } \delta \quad \text{and}$$

$$Y' = Y \text{ Cos } \delta - X \text{ Sin } \delta \quad ,$$

it can be seen that to eliminate the Y term in the left half of equation (8), δ must be chosen such that

$$a \text{ Sin } \delta + b \text{ Cos } \delta = 0 \quad , \text{ or}$$

$$a^2 \text{ Sin}^2\delta = b^2 \text{ Cos}^2\delta \quad , \text{ or}$$

$$\text{Sin}^2\delta = \frac{b^2}{a^2 + b^2} \text{ and } \text{Cos}^2\delta = \frac{a^2}{a^2 + b^2}, \text{ or}$$

$$\text{Sin } \delta = \frac{-b}{\sqrt{a^2 + b^2}} \text{ and } \text{Cos } \delta = \frac{a}{\sqrt{a^2 + b^2}}$$

applying this to equation (8) *we have*

$$\frac{a^2X'}{\sqrt{a^2 + b^2}} - \frac{abY'}{\sqrt{a^2 + b^2}} + \frac{abY'}{\sqrt{a^2 + b^2}} + \frac{b^2X'}{\sqrt{a^2 + b^2}} = c\sqrt{(X')^2 + (Y')^2 + Z^2}, \text{ or}$$

$$\frac{a^2 + b^2}{\sqrt{a^2 + b^2}} X' = c\sqrt{(X')^2 + (Y')^2 + Z^2} \quad , \text{ or}$$

$$\frac{X'}{\sqrt{(X')^2 + (Y')^2 + Z^2}} = \frac{c}{\sqrt{a^2 + b^2}}$$

Resubstituting,

$$\frac{X'}{\sqrt{(X')^2 + (Y')^2 + Z^2}} = \frac{\frac{C_0}{V_0}\,(f_R - f_0)}{\sqrt{(f_0 \text{ Cos } \omega t + f_R)^2 + f_0^2 \text{ Sin}^2\, \omega t}} \quad \text{, or}$$

$$\frac{X'}{\sqrt{(X')^2 + (Y')^2 + Z^2}} = \frac{\frac{C_0}{V_0}\,(f_R - f_0)}{\sqrt{f_0^2 + f_R^2 + 2f_R f_0 \text{ Cos } \omega t}}$$

remembering that $\sqrt{(X')^2 + (Y')^2 + Z^2} = C_0 t$, the distance that the sound has traveled in time t.

But the left hand member of this equation is cos γ', the X' direction cosine. For a particular set of values of the parameters in the right-hand member of the equation, cosγ', and hence γ' are constant. Thus it can be seen that for each time, t, the intersection of the iso-frequency surface of frequency f_R and the sphere of radius $C_0 t$ is a circle centered on the X' *axis, which is at angle* δ *from the X axis. This circle defines a spherical* segment.

Let us define our differential of the volume enclosed by an iso-frequency surface as the area of this segment times the differential of range or

$$dv = A\, dr$$

Now the area of a segment with thickness of h of a sphere of radious r is

$$A = 2\pi r h \qquad \text{where}$$

$$h = r - r \text{ Cos } \gamma$$

Therefore

$$A = 2\pi r^2\, (1 - \text{Cos } \gamma) \quad \text{, and}$$

$$dv = 2\pi r^2\, (1 - \text{Cos } \gamma)\, dr$$

Substituting for r, dr, and Cos γ we have

$$dv = 2\pi C_0^3 t^2 \left[1 - \frac{\frac{C_0}{V_0}\,(f_R - f_0)}{\sqrt{f_0^2 + f_R^2 + 2f_R f_0 \text{ Cos } \omega t}} \right] dt$$

The volume returning energy from within the iso-frequency surface at time t will be

$$V = 2\pi C_0^{3} \int_{t - \frac{\Delta t}{2}}^{t + \frac{\Delta t}{2}} t^2 \left[1 - \frac{\frac{C_0}{V_0}(f_R - f_0)}{\sqrt{f_0^{\,2} + f_R^{\,2} + 2 f_R f_0 \cos \omega t}} \right] dt \qquad (9)$$

The integral in equation (9) has no analytic solution, but is evaluated digitally. The volume returning energy within any band, obviously, would be the difference between two such integrals at the two values of f_R representing the band limits.

4.4 Verification

Appendix F gives the results of an extensive comparison between various models for computing the boundary reverberation. This compares results only for straight-running cases. In addition, comparisons were made in 1973 with an independent model developed by Dr. J. H. Slaton of the Naval Undersea Center for a limited number of turning cases. These revealed several difficulties in the turning case which have been corrected. Agreement was very good, considering the difference in mathematical approach.

For the straight-running case, comparisons of volume reverberation with the method described in reference 1 have been excellent. This was expected because of the similarity in assumptions in the two very simplified models.

SECTION V

AREAS FOR POSSIBLE IMPROVEMENTS AND REFINEMENTS

5.1 Volume Reverberation

The simulation of volume reverberation includes two gross simplifications; the medium is homogeneous, and it is unbounded. Thus, several known features of particular environments cannot be realistically approximated. In particular, returns from scattering layers, returns from regions of greater or lesser focusing, e.g. caustics, and returns over multiple paths are ignored. Still it is hoped that the model can supply numbers of the right order of magnitude in some useful cases.

Almost from the beginning, the three-dimensional analogue of the boundary calculations was recognized as more realistic. A preliminary study was made in 1967 by M.M. Jacoby of NUC, which outlined the geometric problems of computing incremental volumes. Although precise formulae have still not been developed, the problems seem mainly those of bookkeeping; that is, of ensuring that the entire insonified volume is accounted for. The calculations would also be very costly in computer time.

Implementations of this model: tedious and time-consuming to write and check-out, but require only re-writing of one DOP subroutine, plus the changes in SONAR mentioned in 5.2.

5.2 Multi-path Reverberation

The current model approximates the ray-path losses when transmit and receive paths are not the same. The approximation is simply the average of the two-way losses over each single path. Now the transmitted and reflected rays have different spreading loss even over the same (reciprocal) path. The difference is small, however, rarely amounting to so much as 0.2 dB.

Another discrepancy is in the implicit averaging of the scattering strengths. The tacit assumption (not to be dignified as an approximation) is that scattering strength from one path into another is the average of the back scattering strengths for the two paths. In the absence of a good theoretical or empirical alternative, this assumption was allowed to stand. There is a need for a single expression of scattering strength in all directions from an incident ray.

Incorporating such an expression in DOP would require that scattering strength be removed from total losses passed by program SONAR.

Assesment of these changes: given the scattering expression, the latter change would be an easy matter; the former would not be much more difficult.

5.3 Vehicle Translation During Ping

Appendix E discusses the consequences of using the same ray-path for transmit and receive, ignoring the translation of the vehicle. This change is of secondary importance, but should probably be included for completeness if the changes in 5.2 were implemented. It would be easy under these circumstances.

5.4 "Spreading" of Reverberation

There are theoretical reasons to believe that doppler shifts caused by motions of scatterers, particularily at the surface, are a function of grazing angle. No such provision is made in this model. Incorporation of this feature would require a fundamental change in the handling of boundary "spreading" and probably minor changes in SONAR and RAYSRT.

Difficulty of these changes: moderately lengthy, but not complicated.

5.5 Preferential Orientation of Vehicle with Respect to the Environment.

Real environments are non-uniform in every direction, not merely in depth as currently modeled.

The following three additions to the model would take account of some non-uniformities. All would come under the heading of major revisions. In addition, it is clear that, conditions being different at all points and times in the environment, a single ping cycle would no longer give a general description of reverberation. The changes would, however, make possible the calculation of ranges of values that might be expected.

5.5.1 Current and/or Wind Direction

Only DOP would need revision. Doppler content due to both vehicle motion and scatterer motion is affected.

5.5.2 Sloping or Broken Bottom

The SONAR program can compute rays reflected from any bottom made of plane segments which reflect rays in the vertical plane of incidence. Such bottom segments must be perpendicular to the plane containing the rays, and such bottoms are of limited usefulness in modeling real environments.

Spreading losses and directions of reflections can be computed for rays striking any analytically describable bottom. However a properly general treatment would require considerable changes not only in SONAR and DOP, but also in RAYSRT. These changes would seem of little value without also the ones in 5.5.3.

5.5.3 Three-dimensional Velocity Profiles

The use of velocity profiles varying in two or three dimensions requires completely new SONAR, RAYSRT and DOP programs; no mere adaptation would suffice. While ray tracing programs already exist and could be adapted, the running times would be increased by an order of magnitude, at least.

REFERENCES

1 - Naval Undersea Center. Digital Computer Programs for Analyzing Acoustic Search Performance in Refractive Waters, by Philip Marsh and A. B. Poynter, Pasadena, California, NUC, DEC 1969 (NUC TP 164, Vols. 1 & 2).

2 - NUC TP 164 Vol. 3 prepared by H. C. Bertuccelli of Bendix Corporation updating Vols. 1 & 2.

3 - The Bendix Corporation. Plotting Program SRNBT4 and Density Spreading Program DENFSP, by H. C. Bertuccelli, re programs of A. B. Poynter, North Hollywood, California, OCT 1975 (Bendix SR 75.58).

INITIAL DISTRIBUTION

Operational Test and Evaluation Force (1)

Deputy, Operational Test and Evaluation Force, Pacific (1)

Naval Sea Systems Command (SEA-662D) (1)
(SEA-662D-21) (1)
(SEA-03B) (1)
(SEA-0342) (3)

Office of Naval Research (ONR-466) (1)

Anti-Submarine Warfare Systems Project (PM-4) (ASW-13) (1)
(ASW-14) (1)
(ASW-54) (1)

National Oceanographic Data Center (1)

Naval Electronics Laboratory Center, San Diego (1)

Naval Ship Research and Development Center
Annapolis Division (1)
Carderock Division (1)

Naval Oceanographic Office (1)

Naval Surface Weapons Center, White Oak (1)

Naval Postgraduate School, Monterey (Library, Technical Reports Section) (1)

Naval Research Laboratory (1)

Naval Torpedo Station, Keyport
Quality Evaluation Laboratory, Technical Library (1)
Director, Research and Engineering (1)

Naval Underwater Systems Center, Newport (1)

Naval Underwater Systems Center, New London (1)

Naval Weapons Center, China Lake (Code 753) (1)

Defense Documentation Center (TISIA-1) (20)

Applied Physics Laboratory, University of Washington, Seattle (1)

Applied Research Laboratory, Pennsylvania State University (1)

Honeywell, Inc., Hopkins (1)

Honeywell, Inc., Seattle (1)

Bendix, Sylmar (1)

Naval Undersea Center

Code 14
252
3044
307
35021
353
35302
3542
3543
3573 (3)
401
4532
4534 (2)
6565

BELT,IL A.TEST-PROBLEM/DOP-SONARS
ELTOT7 RLIB70 07/23-12:45:39-(,0)

000001	000	1REVERB	16.	2.	1800.	-31.4159		-60.	15.	
000002	000	2								
000003	000	36.	7.	8.	9.					
000004	000	4-90.	1.5	0.	-31.4	-90.	9.	0.	0.	
000005	000	4-82.	-1.5	10.	-25.9	0.	9.	8.	19.8	
000006	000	4-77.	-5.	20.	-22.3	0.		10.	22.9	
000007	000	4-68.	-16.5	30.	-19.6			12.	25.3	
000008	000	4-60.	-24.	40.	-17.2			14.	27.1	
000009	000	4-53.	-28.	50.	-14.9			16.	28.3	
000010	000	4-48.	-30.	60.	-12.9			18.	29.3	
000011	000	4-40.	-32.	70.	-11.1			20.	30.1	
000012	000	4-30.	-34.	80.	-9.2			25.	31.4	
000013	000	4-9.	-37.	90.	-7.6			30.	32.	
000014	000	40.	-43.	90.				40.	31.6	
000015	000	40.						50.	27.4	
000016	000	D						90.	22.8	
000017	000	31.	2.	3.	5.					
000018	000	4 -180.	0.	0.	0.	0.	0.	0.	0.	
000019	000	D 180.	0.	10.	0.	200.	0.	200.	0.	
000020	000	3 4.	10.							
000021	000	4 0.	0.	0.	4000.					
000022	000	D 10.	0.	9000.	4000.					
000023	000	C11.	12.							
000024	000	50.		2820421.4		-.27303767		69.502715		3.70
000025	000	525.								
000026	000	550.								
000027	000	5100.								
000028	000	5150.								
000029	000	5180.								
000030	000	5190.								
000031	000	5200.								3.70
000032	000	5205.								
000033	000	5208.50815		2820421.4		-.27303767		69.502715		
000034	000	5215.								
000035	000	5225.								
000036	000	5235.								
000037	000	5245.								
000038	000	5250.00149		2820409.3		-.29729444		70.631276		3.71
000039	000	5265.								
000040	000	5280.								
000041	000	5284.68723		2819915.8		-3610.669		94.093745		
000042	000	5284.85								
000043	000	5285.00194		1840654.5		0.18280022		1473.2444		3.72
000044	000	5320.								
000045	000	5350.								4.31
000046	000	5380.								
000047	000	5403.								4.78
000048	000	5430.								
000049	000	5450.00484		-1364724.5		0.021384351		-8236.7312		5.16
000050	000	5470.								
000051	000	5491.16777		2680549.1		811.65644		164.15731		
000052	000	5492.00578		4796782.1		-.0645145[illegible]3		-3334.0923		5.50
000053	000	5520.								
000054	000	5550.								5.68
000055	000	5570.								

APPENDIX A

FIGURE 1

```
000056     000   5600.00860      4201878.8     -.068559745     -2713.1275      5.80
000057     000   5630.
000058     000   5655.968408     2643857.5     8233.5634       218.67150
000059     000   5656.01029      3669419.3     -.00065898565   -24043.178      6.00
000060     000   5680.
000061     000   5700.                                                         6.02
000062     000   5720.
000063     000   5750.01344      2640912.4     0.33833872      284.05826
000064     000   5765.
000065     000   5783.34800      2641521.0     -1.1518146      254.37530
000066     000   5792.
000067     000   5800.01529      2627055.9     0.037415208     643.14143       6.08
000068     000   5850.
000069     000   5900.                                                         6.12
000070     000   5950.
000071     000   51000.                                                        6.18
000072     000   51100.
000073     000   51250.
000074     000   51312.0411      2629880.5     0.11624671      503.65531       6.32
000075     000   51400.
000076     000   51510.9659      2629880.5     0.11624671      503.65531
000077     000   51537.3904      2629909.7     -.49887228      514.51595
000078     000   51543.5479      2629909.7     -.49887228      514.51595
000079     000   51650.0650      2618522.7     0.0839802       760.02509       6.41
000080     000   51900.
000081     000   51968.                                                        6.58
000082     000   52280.0753      2618522.7     0.0839802       760.02509
000083     000   52297.                                                        6.71
000084     000   52400.
000085     000   52500.1493      2622869.3     -.031541745     1029.0237       6.72
000086     000   52790.8537      2621947.8     0.22082943      916.17354
000087     000   52936.2060      2670552.4     -.011193266     2234.7444       6.79
000088     000   53500.                                                        6.77
000089     000   53904.                                                        6.76
000090     000   54000.                                                        6.77
000091     000   6DOPLER   750.     119.     17.     -2.     2.9        020040
000092     000   7-90.          90.            1.
000093     000   7              MAXN
000094     000   7              MAXP
000095     000   7              DEPTHN
000096     000   7              DEPTHP
000097     000   +
000098     000   6DOPL 2   500.     119.     17.     -2.     2.9        020040
000099     000   GLIST
000100     000   6DOPL 3   275.     119.     17.     -2.     2.9        020040
000101     000   +
000102     000   GLIST
000103     000   .

END ELT.
```

APPENDIX A

FIGURE 1 (CONTINUED)

FIGURE 2

```
@       ELT,IL    A.TEST-PROBLEM/DOP-RAYSRT
ELTOT7 RLIB70 09/24-14:39:18-(,0)
000001        000          PLOT
000002        000   0.          4000.
000003        000   +
000004        000   0.          4000.
000005        000   +
000006        000   0.          4000.
000007        000   +

END ELT.
```

FIGURE 3

```
@ELT,IL A.TEST-PROBLEM/DOP-DOP
ELTOT7 RLIB70 07/23-12:45:50-(,0)
000001        000   FO = 3.1415926535 E1
000002        000   VS= 45., BWIDTH = .25, DELT = .040, TIME =
000003        000   1.5       ,2.0
000004        000   1.5       ,2.0
000005        000   .250
000006        000    THTMAX = 30.
000007        000   BWIDTH = 5.
000008        000   GO
000009        000   THTMAX = 180.
000010        000   KNOTS,RELATIVE
000011        000   OMEGA = 10., NBEAM = 1
000012        000   CENTER
000013        000   FILTER
000014        000   VSPRED=.2,.1,.1,.1,.1
000015        000   SSPRED = .5,.125,.0625,.03125,.015625,.015625
000016        000   BSPRED=.5,.2,.05
000017        000   BWIDTH = 1.
000018        000   PRINT EVERY = 5
000019        000     TIME = .750
000020        000   GO
000021        000   TIMECOMP, TOTALS
000022        000   TVG
000023        000          PLOT
000024        000   GO
000025        000   GO, END

END ELT.
```

APPENDIX A (CONTINUED)

APPENDIX B

MEMORANDUM

P80203/ABP:pas
18 January 1965

From: Code P80203
To: Code P802

Subject: Error from Using 10 $\log_{10}\tau$ as a Measure of Effective Train Length When Computing Boundary Reverberation.

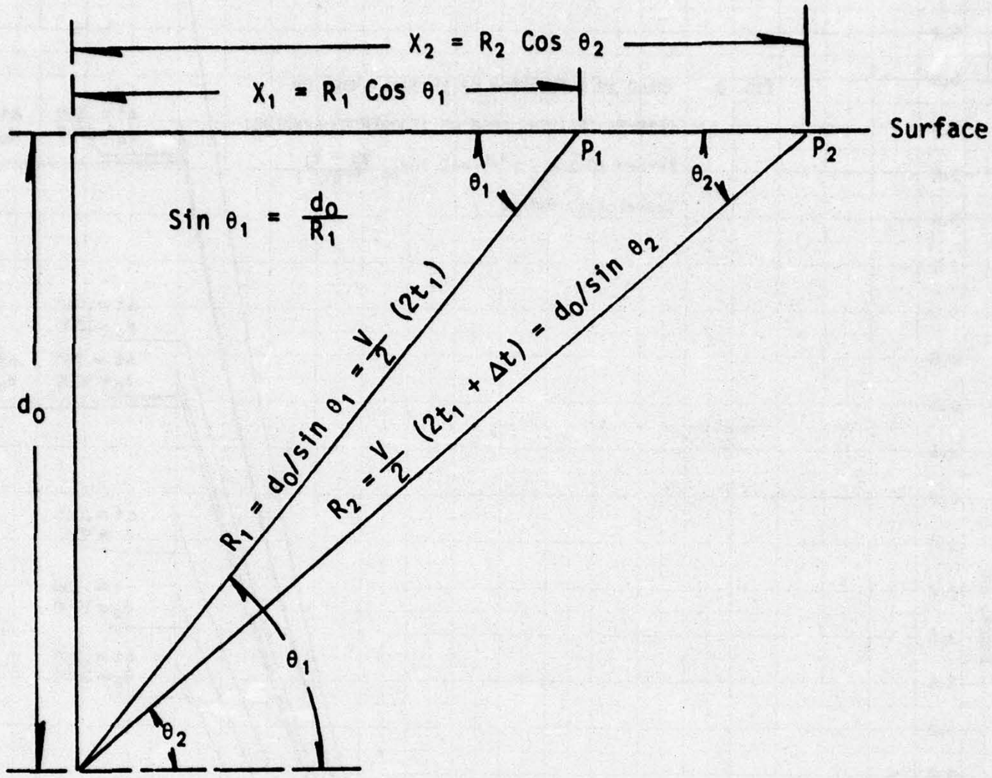
In mathematical models commonly in use for computing boundary reverberation levels in dB, a factor of 10 $\log_{10}\tau$ is added in to account for the fact that an insonified annulus of finite width returns energy to the transducer at the same instant of time. In this context τ is taken to be $V(\Delta t)/2$ where V is the velocity of sound and Δt is the ping length in seconds. It is rather obvious that τ is an accurate measure of annulus width only when the transmission path is essentially horizontal. Since analyses are contemplated for applications in which long pings and steep paths are involved, it seemed desirable to investigate the magnitude of errors which might accrue.

Some simple calculations were made after the fashion indicated in Figure 1, assuming an isotropic medium (V = 5000 ft/sec). Starting with a source depth and an initial ray angle relative to the horizontal, the slant range (R_1) to surface intercept was computed along with the horizontal range (X_1) to this point and the two-way travel time. If time is measured from the end of transmission, this two-way travel time ($2t_1$) would identify the instant when the trailing edge of the wave train will return energy from point P_1. At this same instant the leading edge of the wave train will be returning energy from point (P_2) for which the horizontal range (X_2) can be found on the basis of a slant range (R_2) consistent with a two-way travel time of $2t_2 = 2t_1 + \Delta t$. The width of the insonified annulus is $\tau^* = X_2 - X_1$. A measure of the dB error involved by using τ instead of τ^* is given by 10 $\log_{10} \frac{\tau^*}{\tau}$.

Such calculations were carried out over a range of values for Q_1 of 20° through 80° for combinations of conditions that can be obtained from source depths of 500, 1000, 2000, and 5000 feet, and ping lengths of .040, .100, and .250 seconds. The results are plotted in Figure 2. The error as plotted is the number of dB that should be added to values computed when 10 log τ is used. It appears that the error from this source is negligible when the significant paths are less than 20° from the horizontal as is the case for the directional systems we have previously analyzed in circular search. Even when a refractive environment is considered, the uncertainty in using an average velocity in computing τ should not increase the error appreciably. As expected, the inaccuracy mounts rapidly with increasing path angles above 20° and with increasing source depth. However, the decrease in error as the ping length becomes longer was not anticipated. This trend occurs because in the ratio $\frac{\tau^*}{\tau}$ the numerator increases less rapidly than does the denominator with increasing ping length.

If the annulus width itself were the only potential source of error, it would be possible to devise a correction procedure. Unfortunately, the contribution of each unit width of the insonified area to the reverberation level is not necessarily the same since the transmission and vertical pattern losses may vary substantially for the different paths involved. Nor in general can a total error be computed by summing the independent errors for each of the factors handled separately. A method for assessing the composite error will be presented in a subsequent memorandum.

A. B. POYNTER



$$P_2P_1 = R_1^2 + R_2^2 - 2R_1R_2 \cos(\theta_1 - \theta_2)$$

FIGURE 1

Method for finding $\tau^* = X_2 - X_1$

OD 52258

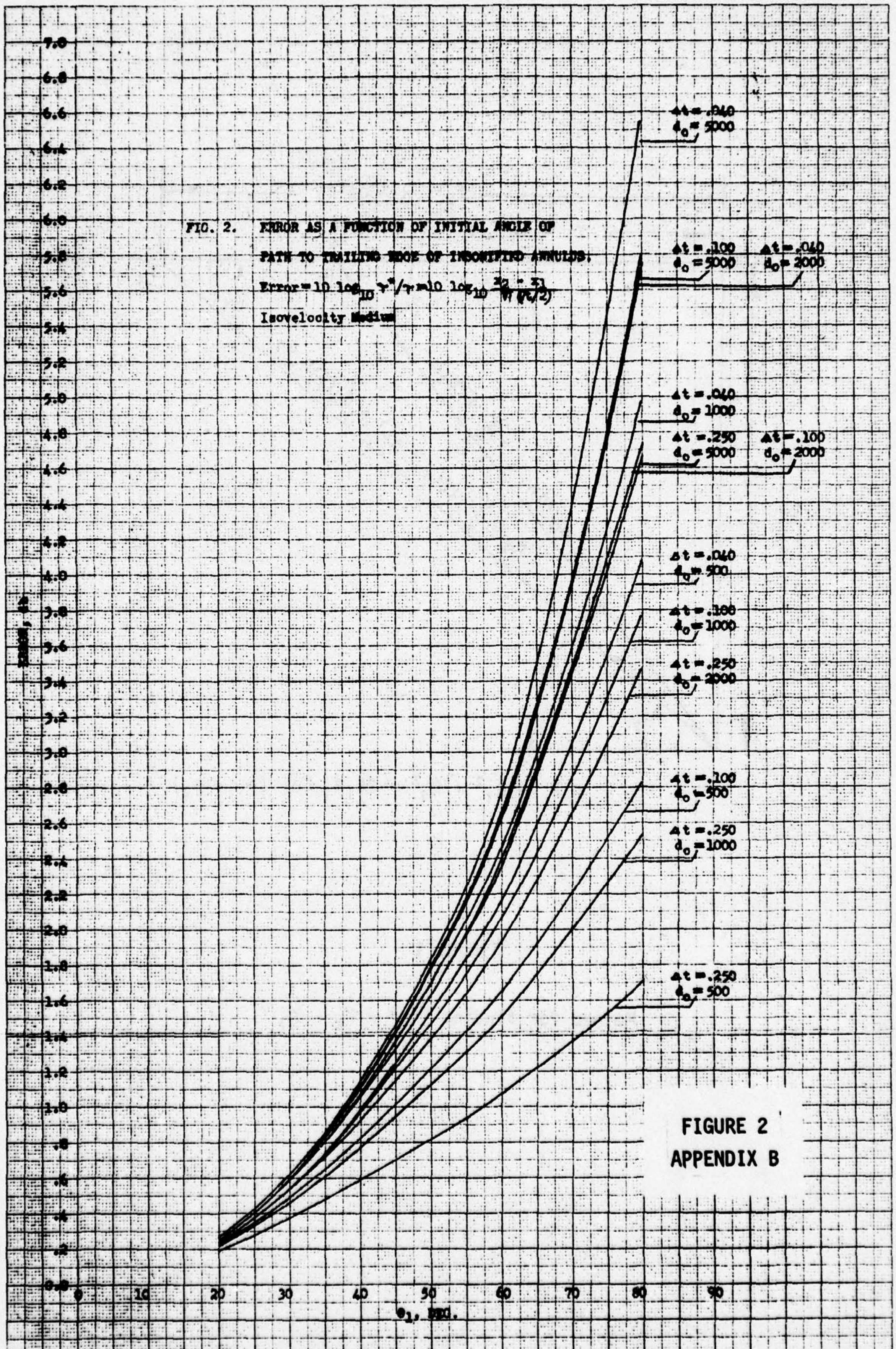


FIGURE 2
APPENDIX B

Fig. 2.

APPENDIX C

MEMORANDUM

P80203/ABP:pas
20 January 1965

From: Code P80203
To: Code P802

Subject: Review of Mathematical Model for Computing Boundary Reverberation.

References: A. NAVORD Report 5606 (NOTS 1818) - Analytical Methods for Predicting Acoustic Performance of Homing Torpedoes in Circular Search.

B. TPR 246 (NOTS TP 2498) - Analytical Studies of Sonar in Refractive Water.

C. P80203 Memo, "Error from Using 10 $\log_{10}\tau$ as a Measure of Effective Train Length when Computing Boundary Reverberation," of 18 January 1965.

In the analytical work, it is necessary to develop mathematical models of the phenomenon to be studied so that quantitative results can be computed. These models are seldom exact, either because the process is not completely understood or because of a desire to avoid complexities which do not materially affect the results. The models may be adequate for the purpose originally intended, but it is dangerous to extend their use to new situations without reassessing their adequacy. The surface and bottom reverberation equations used in our computer programs are cases in point. The equations in general use are those given in References A and B.

For boundary reverberation a ray is traced to the surface (or bottom) and the expected reverberation level at the time in the ping cycle corresponding to the two-way travel time is computed, using: the transmission loss over this path; vertical pattern losses based on the initial ray angle, and a scattering coefficient per unit area based on the grazing angle at which the ray strikes the surface. Since the annular ring on the surface which returns energy at the time in question is unlikely to be unit distance wide, an additional term ($10\log_{10}\tau$) is added (τ = half the train length = $V(\Delta t)/2$ where V is the velocity of sound and Δt is the ping duration in seconds). The equation was formulated in the context of fairly shallow systems using relatively short pings and highly directional transducers oriented to emphasize horizontal coverage. For such systems the equation could be expected to yield acceptable accuracy. Future applications may involve one or more of the following:

- Long ping lengths for correlation detection
- Broad vertical patterns to provide greater depth coverage
- Steep paths (deep source or bottom bounce)

In preparation for such work it seems advisable to re-examine the mathematical model to ascertain whether or not it is still satisfactory.

In Reference C it was shown that 10 $\log_{10}\tau$ may not adequately account for the linear extent of the insonified area of the boundary which returns energy at the same instant of time. Moreover, when this annulus has considerable width, as is the case for the longer pings, the contribution of various parts of the insonified area to the total energy return may be substantially different because of variations in transmission loss, pattern loss, and even in the scattering coefficient for the different paths involved. To better account for these things and to provide a reference for assessing the accuracy of the model now in use, modifications were made in the computational procedures.

The key to the method is the temporary assumption that 10 $\log_{10}\tau = 0$. This is equivalent to saying that a ping length is chosen so that the insonified annulus at the boundary is 1-yard wide. This is consistent with the way in which the scattering coefficient is given as a function of grazing angle, and, over such a distance, no change in transmission or pattern loss need be considered. Otherwise, the desired sets of environmental conditions and equipment parameters are used. A whole family of rays are run from the selected source depth to the surface, and reverberation levels are computed as before. However, instead of plotting reverberation level versus two-way travel time as is usual, both reverberation and travel time are plotted against the horizontal range at which each ray strikes the surface. If the beginning of transmission is taken as zero time, any selected time ($2t_i$) on the two-way time curve will identify the horizontal range at the surface from which the leading edge of the wave train is returning reverberation. It follows that $2t_i - \Delta t$ identifies the range from which the trailing edge is returning energy at the same instant. Integrating the reverberation curve between these two limits should give the actual reverberation level quite accurately. It should be noted that the values in the one-yard increments must be converted from dB to intensity before summing and then reconverted. Reasonable accuracy can be achieved with much less work by using average values for range increments several yards wide and multiplying each average value by the number of yards in the increment before summing.

After sufficient points have been determined in this manner, the values of reverberation can be replotted against elapsed time to yield the type of presentation generally desired. To obtain comparable results from the original model, one can rerun the computer program using the appropriate value for 10 $\log_{10}\tau$ as determined by the ping length of the transmission. It is more efficient, however, to return to the working curves and add 10 $\log_{10}\tau$ to the values of the reverberation curve for $\tau = 1$ at the range determined by the travel time curve at the elapsed times for which data are desired.

Surface reverberation levels were computed by the two methods using parameters for the Torpedo MK 46 Mod 0 in circular search. Attenuation by the RRF was not considered. The source depth was assumed to be 750 feet. The velocity profile for the environment is shown in Figure 1A. The new computer program which permits use of a continuous gradient profile was used. The comparative results for a 40-ms ping are shown in Figure 2. The vertical patterns, of course, dominate the shape of the curves at times less than one second. The integration procedure would normally be expected to lower the peaks and fill in the troughs as well as to shift them slightly. The fact that the peaks in this case are lower for the curve obtained via the old

model suggests that the 10 log τ correction is insufficient, particularly at the shorter times which would be identified with steeper paths. It might be kept in mind that the curve obtained by means of the original equation has the same shape as would the 10 log τ = 0 curve. The initial ray angles associated with a few points on this curve are shown as a matter of interest.

Figure 3 shows the comparative reverberation levels that would obtain if a ping length of 250-ms were used. It is clear that one must be very careful in computing reverberation levels when long pings are used; the two curves are substantially different. The integrated curve was not determined at times less than .55 seconds because before that time all of the ping has not reached the surface.

For both ping lengths, the curves would match more closely if tl curves obtained by means of the original model were advanced in time by one-half the ping length. This follows since the computed ray path would then, in effect, intercept the surface near the middle of the insonified area and the transmission loss, vertical pattern loss, and scattering coefficient would be more representative of the area as a whole than is the case when the ray falls at one end of the insonified area. Figures 4 and 5 compare the results when this subterfuge is used for the .040 and .250 ping lengths, respectively.

As a further check, additional calculations were made for a non-turning vehicle at 500 feet in an environment characterized by velocity Profile B shown in Figure 1. Our old computer program based on linear-gradient layers was used in this instance. Other parameters were unchanged. The results are similar to those from the first set of calculations. Figure 6 compares the 40-ms ping data from the integrated method with that from the original model after the latter had been advanced in time by half the ping length.

Figure 7 presents similar data for the 250-ms ping. Part A, on the left shows the results out to 1.4 seconds. Calculations for this case were carried out to about 7.5 seconds, and the curves (after translation) remain in close agreement until about 6.7 seconds. For this elapsed time the paths involved are from rays that start out below the horizontal and are refracted back toward the surface by the positive velocity gradients which characterize this environment below 170 feet. The ray having a two-way travel time of 6.705 seconds and subsequent ones having slightly steeper initial angles, reverse at depths where the velocity gradient becomes less positive. This results in an abrupt increase in transmission loss and a resulting drop in reverberation level as indicated in Figure 7B (recall that the curve based on the original model has been advanced by 0.125 seconds.) In the vicinity of such anomalies the integrated method offers the only approach for computing expected values of boundary reverberation levels with reasonable accuracy.

Some general observations appear warranted:

- Within the limits imposed by the input data, accurate expected levels of surface reverberation as a function of elapsed time can be computed by means of ray tracing

programs by summing the contributions of 1-yard increments of the insonified annulus which contribute at the same instant.

- At present, this can be accomplished manually, but it may be possible to program the computer to this end.

- The present mathematical model can yield results which are reasonably accurate over the elapsed time interval during which the sound paths are such that 10 $\log_{10}\tau$ is a good measure of the width of the insonified annulus and the change in transmission loss, pattern losses, etc. over this annulus is not appreciable.

- The accuracy attainable and/or the time interval over which a given accuracy can be maintained will be increased if, in effect, the elapsed time is measured from the middle of the transmittal ping. If time is measured from the beginning of transmission the computed two-way travel time to the surface for each ray should be increased by $\Delta t/2$ to obtain the time consistent with the computed reverberation level.

- Even if the technique in the above paragraph is used, errors may become substantial when:

    - The paths are very steep so that 10 $\log_{10}\tau$ is not a good measure of annulus width;

    - The insonified annulus is wide enough to encompass a spread of paths through the minor lobes of the transducer;

    - In the vicinity of transmission loss anomalies (either substantial focusing or defocusing)

The above observations should also apply to bottom reverberation, except that additional complications may arise if the bottom is irregular.

A. B. POYNTER

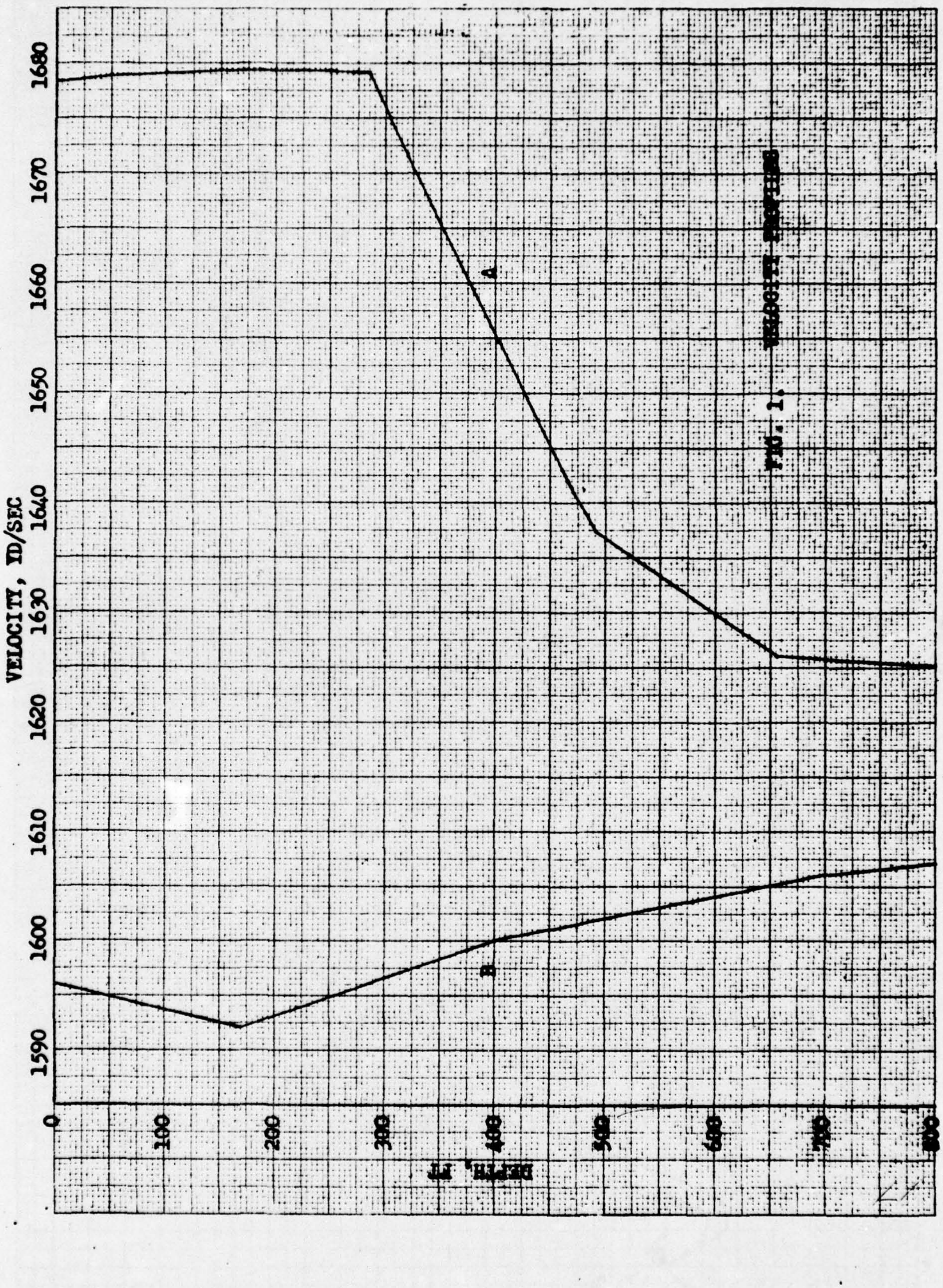


Fig. 1.

APPENDIX C

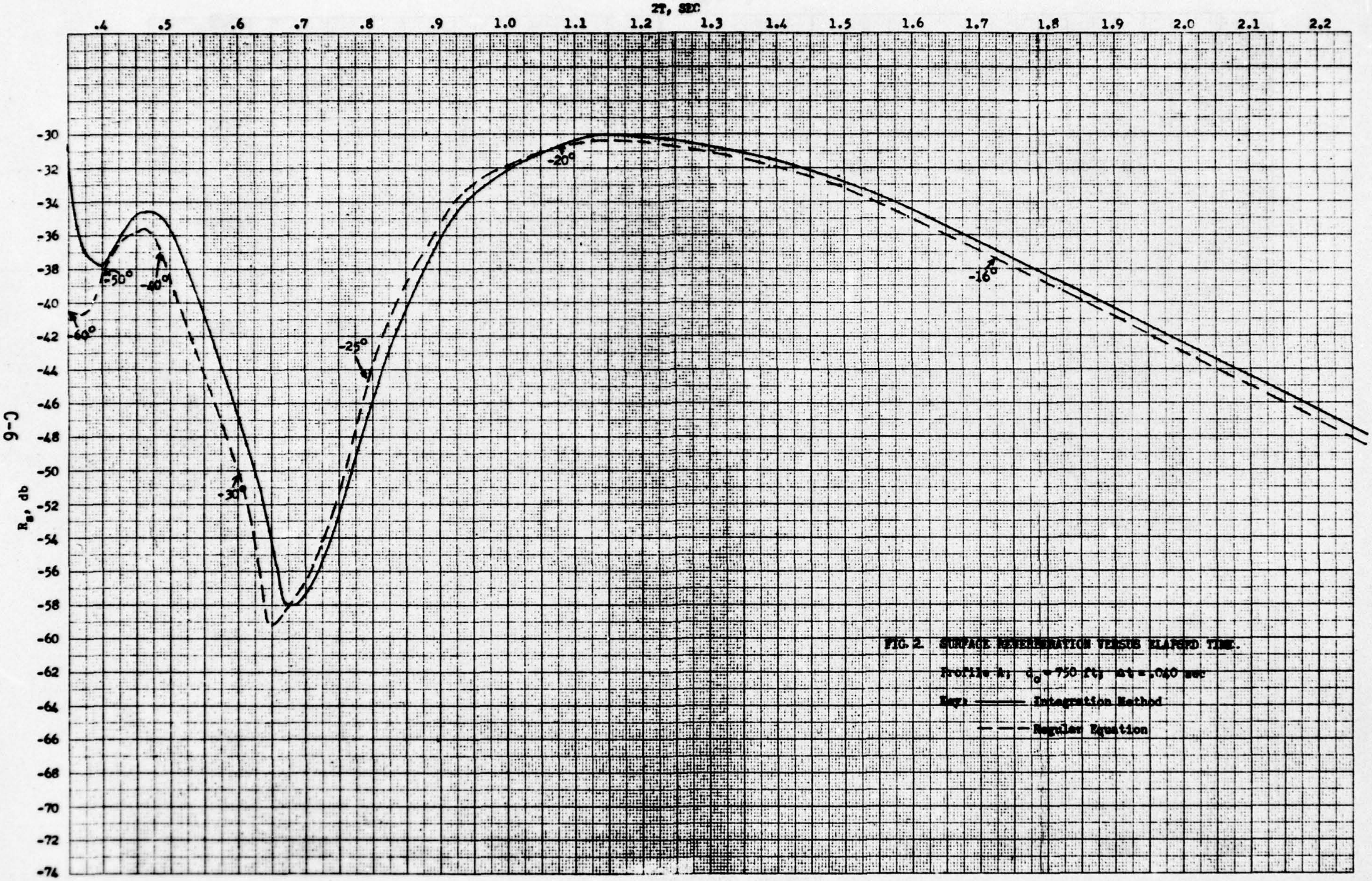


FIG. 2 SURFACE REVERBERATION VERSUS ELAPSED TIME.

Profile A; d_o = 750 ft; Δt = .040 sec

Key: ——— Integration Method

– – – Regular Equation

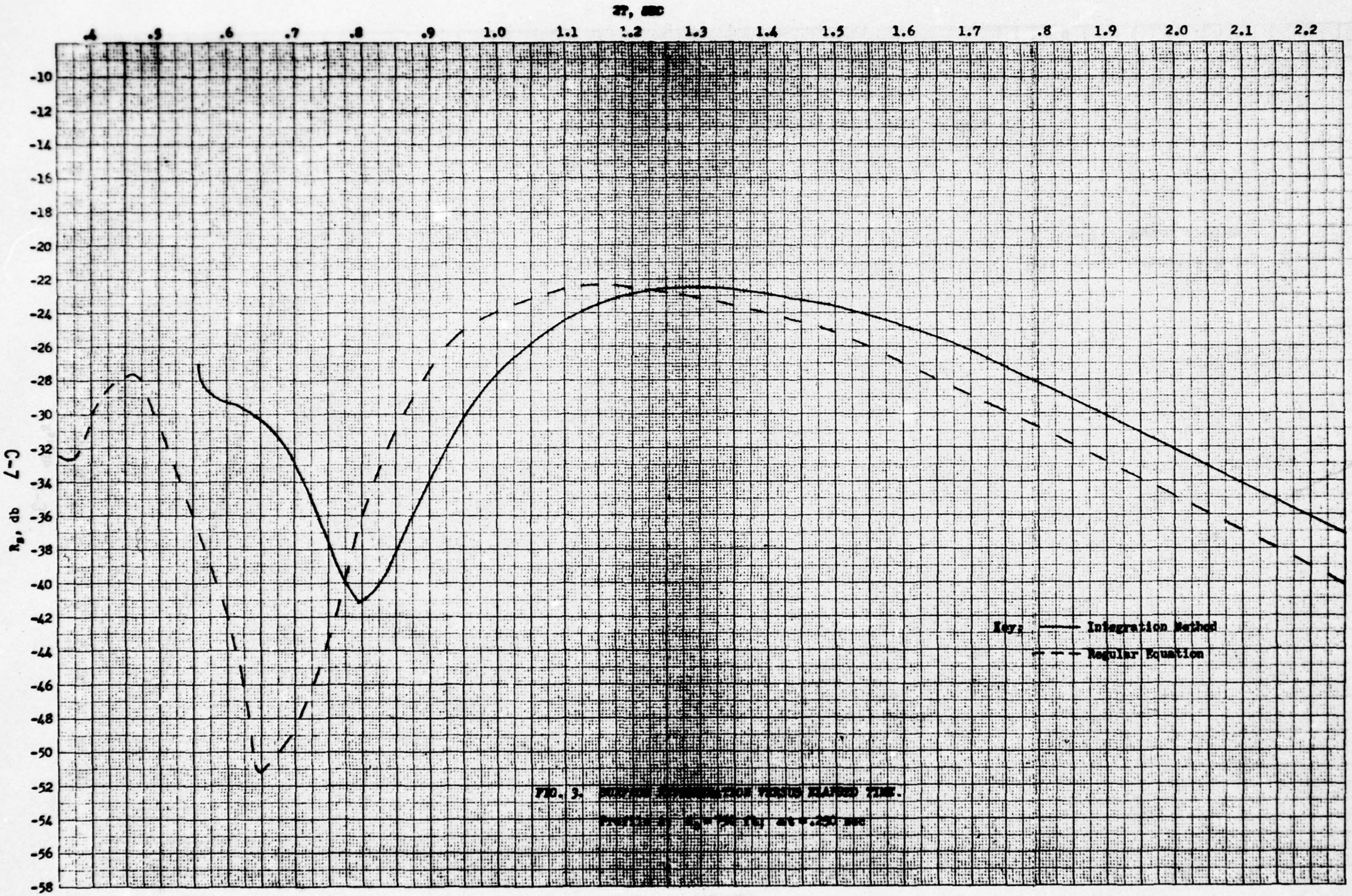


FIG. 3. [illegible] VERSUS ELAPSED TIME.

[illegible]

C-7

APPENDIX C

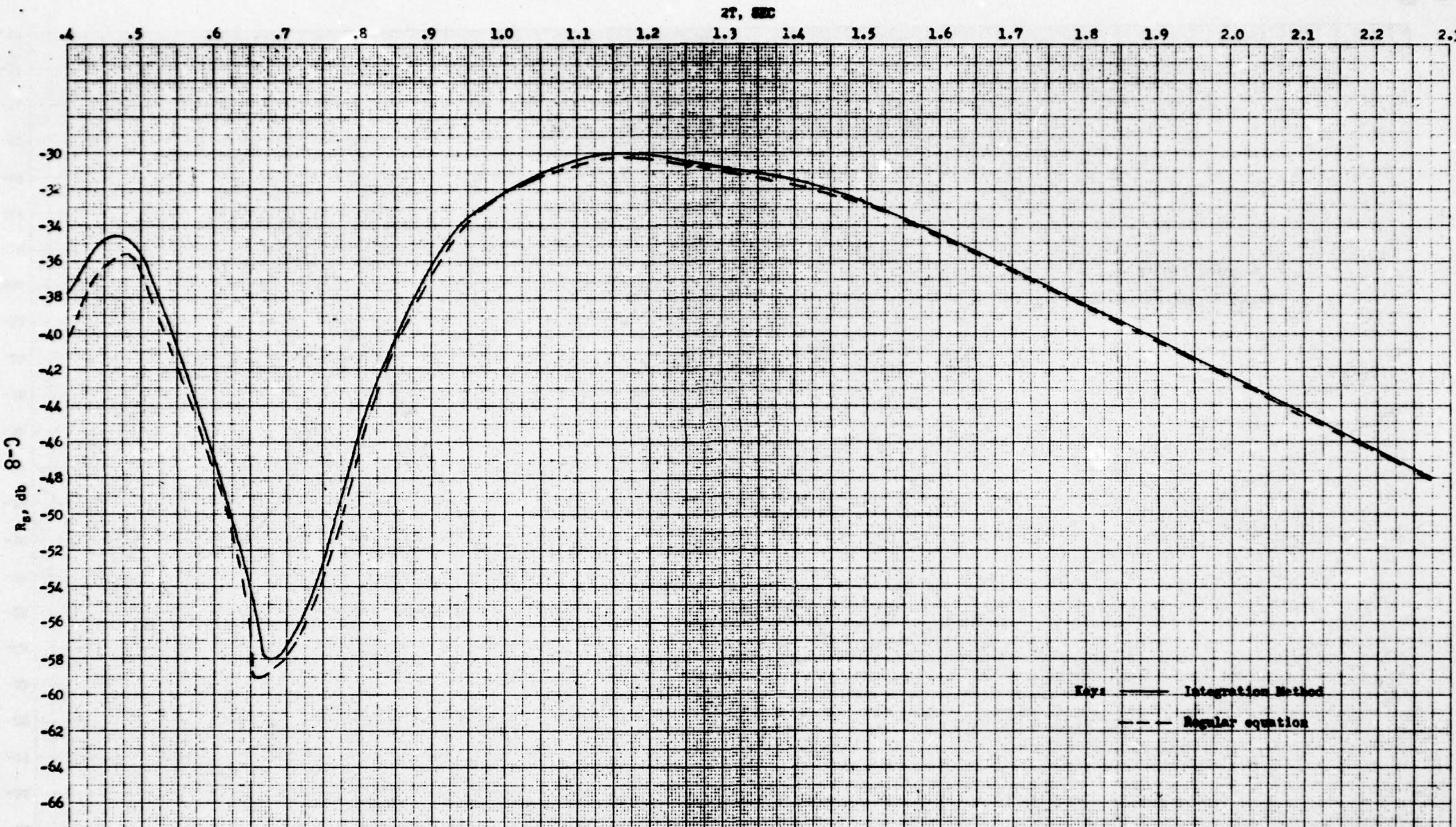


FIG. 4. COMPARISON OF REVERBERATION LEVELS AFTER SHIFTING CURVE OBTAINED BY REGULAR EQUATION BY $\Delta t/2$.

Profile A: d_o = [illegible] ft; Δt = .040 sec

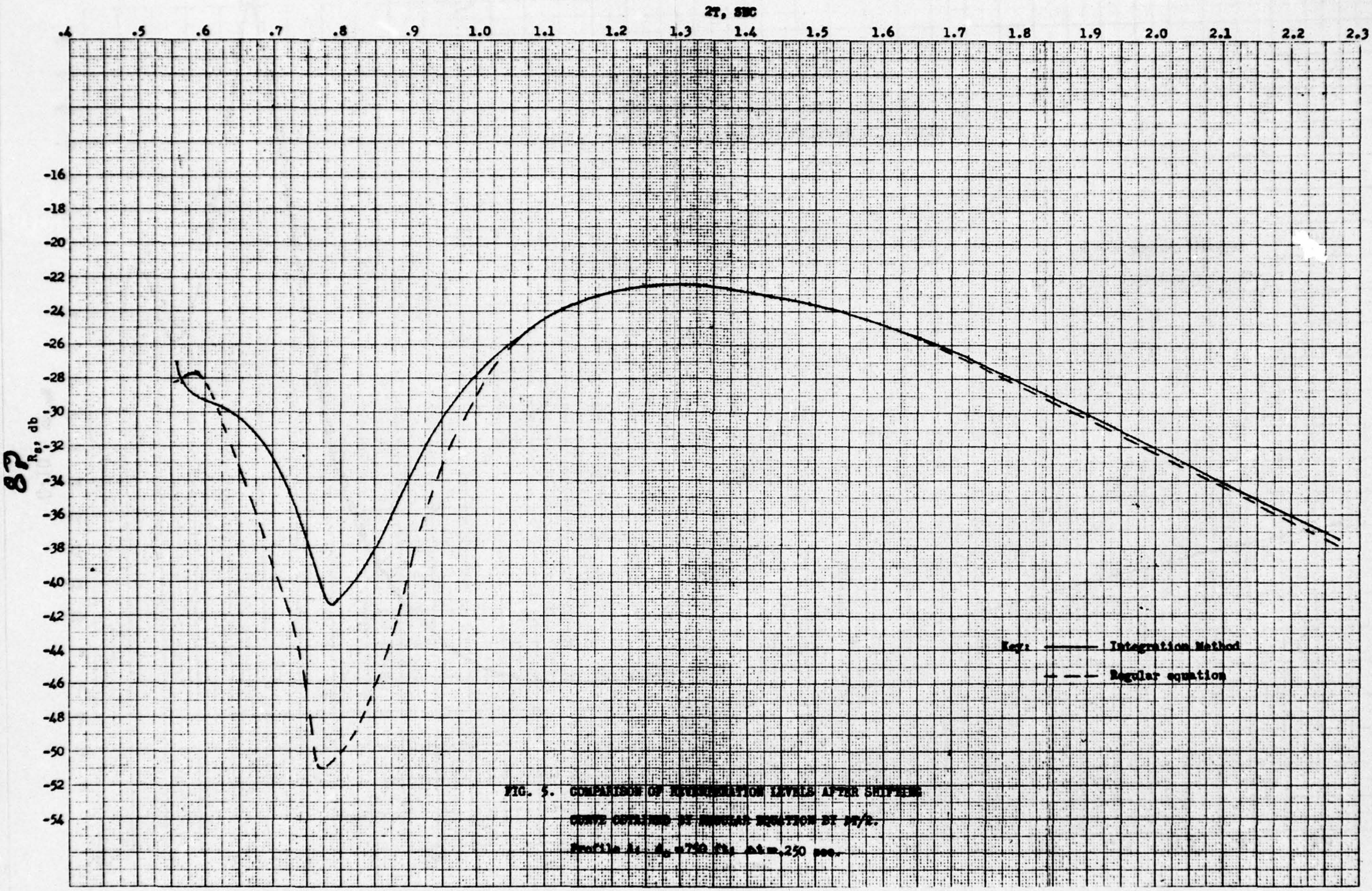


FIG. 5. COMPARISON OF REVERBERATION LEVELS AFTER SHIFTING CURVE OBTAINED BY REGULAR EQUATION BY ΔT/2. Profile A; d_s = 750 ft; Δt = .250 sec.

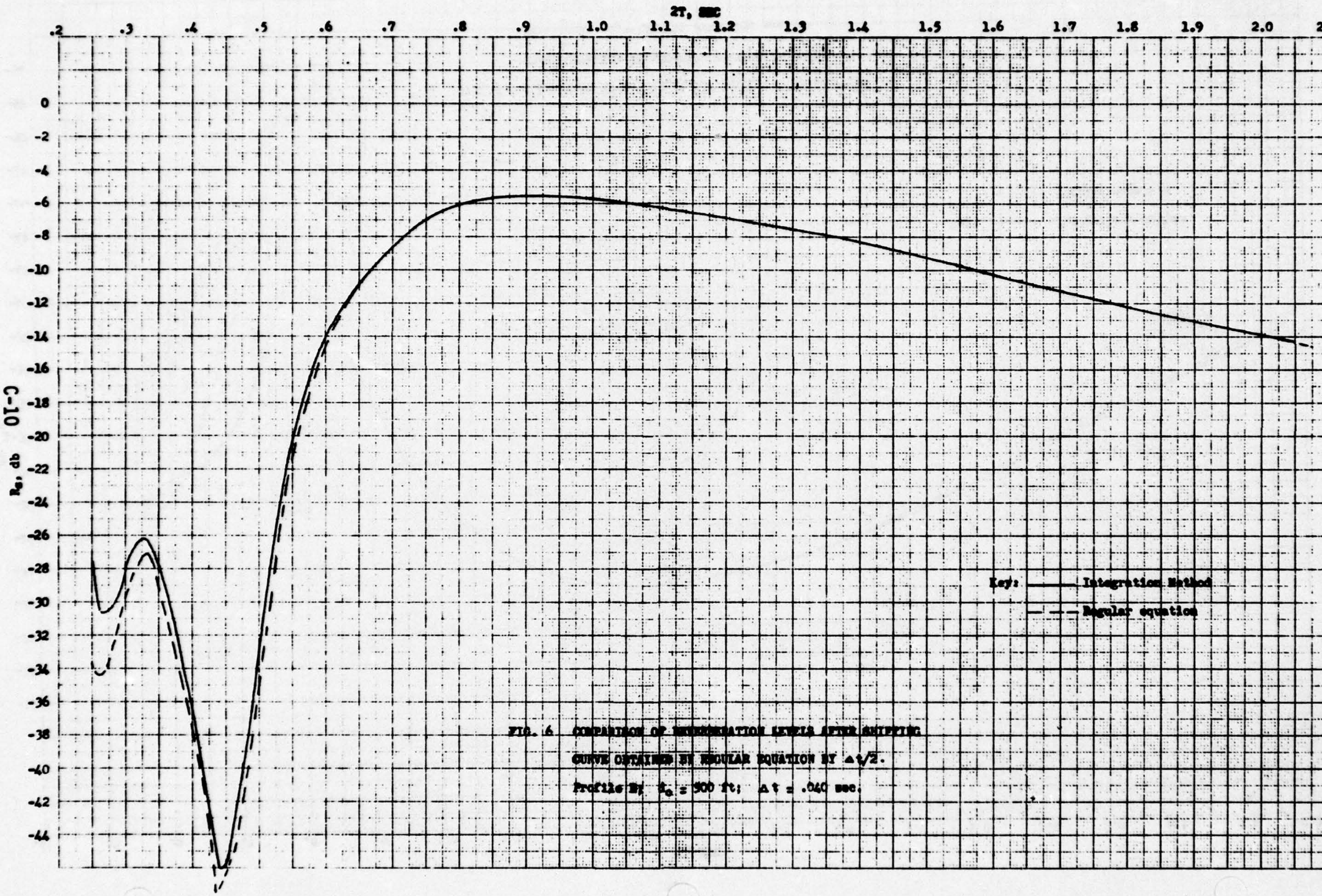


FIG. 6 COMPARISON OF REVERBERATION LEVELS AFTER SHIFTING CURVE OBTAINED BY REGULAR EQUATION BY $\Delta t/2$.

Profile B; z_o = 900 ft; Δt = .040 sec.

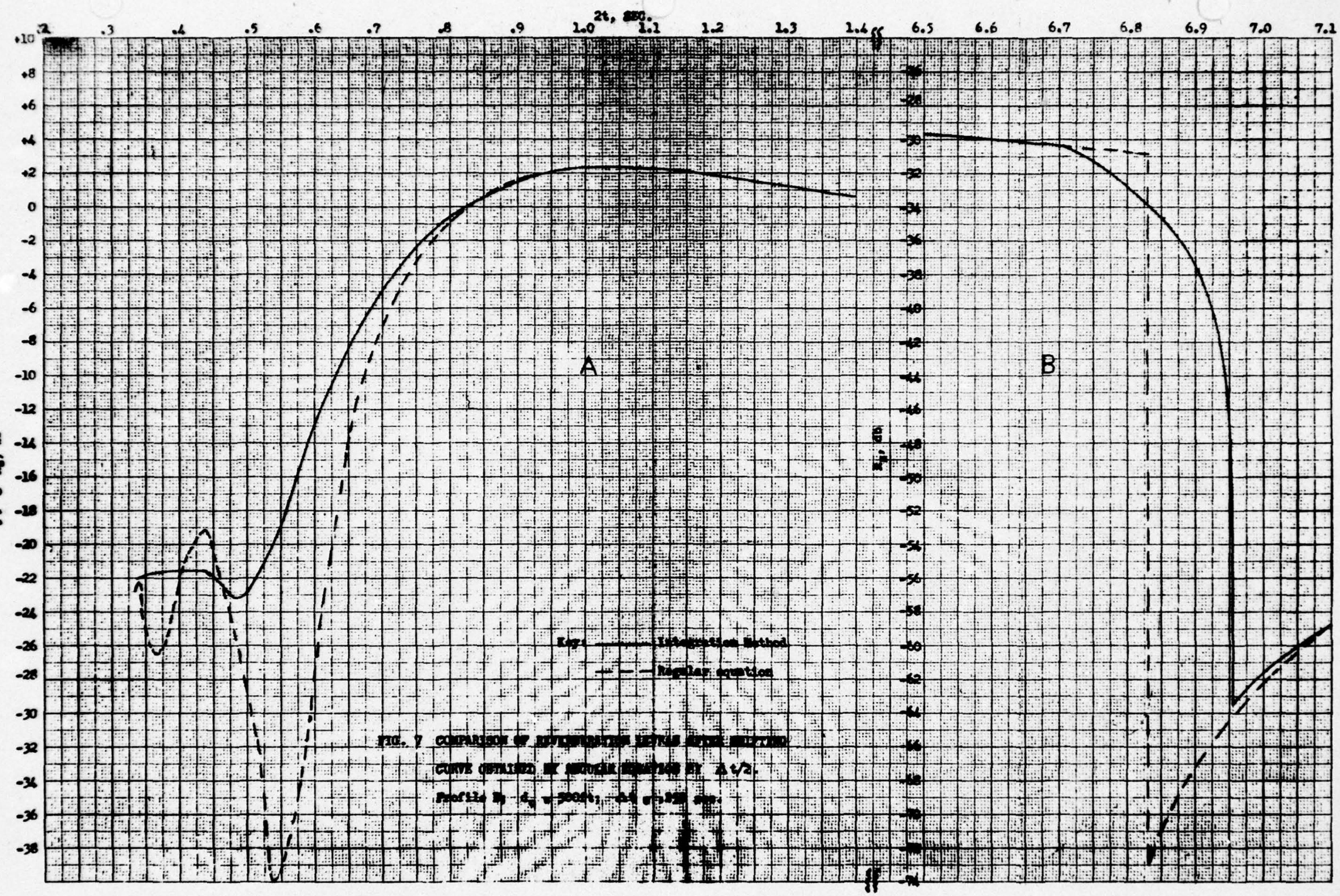


FIG. 7 COMPARISON OF REVERBERATION LEVELS AFTER SHIFTING CURVE OBTAINED BY REGULAR EQUATION BY $\Delta t/2$.
Profile B; d_w = 3000ft; [illegible]

APPENDIX C

OD 52258

APPENDIX D

MEMORANDUM

P80203/ABP:pas
25 January 1965

From: Code P80203
To: Code P802

Subject: Review of Mathematical Model for Computing the Expected Level of Volume Reverberation.

Reference: A. P80203 Memo "Review of Mathematical Model for Computing Boundary Reverberation," of 20 January 1965.

Reference A discussed the problems associated with the mathematical model for computing boundary reverberation. It was pointed out that the model currently in use in our computer programs generally gives reasonable accuracy over a large portion of the ping cycle provided time is measured from the middle of the transmittal ping. The errors may still be appreciable over the time intervals during which the paths to the surface are quite steep and involve transmission through the minor lobe structure of the transducer, and/or exhibit anomalous transmission loss. A method is delineated by which errors can be reduced by summing the contributions from narrow increments of the insonified annulus.

The model currently in use for calculating volume reverberation in dB again involves the addition of a factor $10 \log_{10}\tau = 10 \log V (\Delta t/2)$ to account for the lateral extent of the insonified region which returns energy to the transducer at the same instant of time. (V is the velocity of sound and Δt is the ping duration in seconds.) Since for volume reverberation the insonified region is a spherical shell, the lateral extent is measured radially. Thus $10 \log_{10}\tau$ is a reasonably accurate measure of the thickness of this shell providing the value chosen for V is a good average value for the environment being considered. Since in general the velocity is not constant throughout the volume of interest, there are resulting errors. However, the percentage variation in velocity is so slight that the errors from this source can be expected to be less than 0.2 dB, an insignificant amount in view of our inaccurate knowledge of the scattering coefficients which obtain.

A somewhat more significant error can result from the variation in transmission loss over the shell width, particularly for long pings. To obtain a measure of the inaccuracy, calculations of volume reverberation were made for two postulated ping lengths in an environment with an average velocity taken to be 1660 yd/sec. The levels computed with values of $10 \log_{10}\tau$ appropriate for the selected ping lengths were compared with those obtained by a modification of the integration method described in Reference A with $10 \log_{10}\tau$ set at zero. The levels were plotted as functions of elapsed time measured from the beginning of the transmission. The $10 \log_{10}\tau = 0$ curve can be integrated over the ping length in seconds except that the resulting intensity is multiplied by a factor of 0.83 before reconverting to dB to account for the fact that one millisecond is equivalent to an 0.83-yard shell thickness when V has the value previously selected.

The results for ping lengths of 0.040 and 0.250 seconds are shown in Figure 1. As was the case for boundary reverberation, the corresponding curves from the two methods would be in much better agreement if the one for the integrated method were shifted back in time by half the ping length. This is equivalent to measuring elapsed time from the middle of the transmission. When this is done, there is still a residual error at the very short times due to the fact that the spreading loss at short ranges is changing rapidly. The integration method accounts for the fact that the latter portion of the ping is contributing a considerably greater portion of energy to the reverberation level. The errors for short pings do not seem large enough to justify the additional work of using the integration method.

Our analyses in the past have involved systems with ping lengths of .040 seconds or shorter. The elapsed time for both interference and echo levels was taken to be the two-way travel time for the paths being computed. For the short pings this is roughly equivalent to measuring time from the transmission mid-point. Consequently, the results of the analysis can be considered valid insofar as the factors discussed here are concerned.

There is, however, a basic weakness in our volume reverberation model which may be really serious, especially if performance analyses of long-range systems are attempted. Moreover, there is no means at hand for ascertaining what errors might be involved. This situation arises because the present model for volume reverberation is based on the assumption of an insonified shell over expanding in time in an infinite, uniform medium. It is well known that an ocean is neither infinite in extent nor uniform in its acoustic properties. From whatever transducer position the sound paths will, at one time or another, intercept the surface or bottom, and variation in the physical properties of the water results in differences in transmission loss depending on which path is considered. Furthermore, the volume scattering coefficient in general is not constant over any sizable volume of water, and is often observed to be a rather strong function of depth and time of day (deep scattering layer migration). To further complicate the situation, the paths in various directions are weighted in accordance with the three-dimensional transducer patterns.

The difficulty of realistically delineating the distribution of acoustic properties over a large volume coupled with the even greater difficulty of integrating the return from the entire spherical shell has no doubt led to the common use of the simplified model. Its use can be rationalized as follows. While it is true that the transmission loss to various portions of the insonified shell may be substantially different due to refraction, the integration process will average out such effects. Similarly the variation in scattering coefficient will average out in the integration so that selection of a constant value consistent with the volume being considered should suffice. When the insonified spherical shell is truncated by a boundary, energy aside from the backscattered as boundary reverberation or lost in the reflection mechanism will be returned to the water to insonify volume scatterers. If one can assume that boundary losses are offset by the multiple paths that are set up, the truncating of the sphere can be ignored.

There is little doubt that in a qualitative sense these trends can be expected. There is no available evidence, however, that they quantitatively balance out within acceptable limits. Faith that they will do so deteriorates further when one considers that directional transducer patterns may weight the return from anomalous regions so that their contribution is grossly out of proportion to their share of the total volume. The depth and orientation of the transducer as well as the patterns themselves can be significant factors. For example, if there is a pronounced deep scattering layer, the behavior of the volume reverberation level with elapsed time can be expected to be quite different when a transducer is oriented vertically than when it is oriented horizontally.

One can also argue that the need for a more complex and realistic model for estimating volume reverberation is not great on the basis that most systems have TVG thresholds which protect them against volume reverberation at short range, boundary reverberation is likely to exceed volume reverberation at intermediate ranges, and noise will be the dominant interference at long ranges. While this reasoning may be valid for many (if not most) situations analyzed to date, there is no assurance that it will always hold. Volume scattering coefficients above average when coupled with long pings may raise this type of interference to new prominence in some important situations. The point is, no one can be sure that the simple model is adequate in new applications unless there is a comprehensive model available with which to compare results. Efforts by able mathematical physicists to develop a more realistic approach to the problem of estimating volume reverberation should be solicited. A companion problem of great difficulty is the development of a practical method for estimating the doppler spectrum of both boundary and volume reverberation. Such information is needed for assessing performance of systems which employ doppler discrimination.

In the meantime, the best that one can do is to continue using the present method, taking care that the elapsed time is measured from the middle of the transmitted ping and then adjusted, if desired, to the same time base used in computing boundary reverberation and echo level. It is clear that the expected values of both the echo level and threshold must be figured on the same time base for valid results. It is worth noting that for elongated echoes (long pings or other reasons) the applicable threshold level may vary appreciably depending on which portion of the echo leads to the detection.

A. B. POYNTER

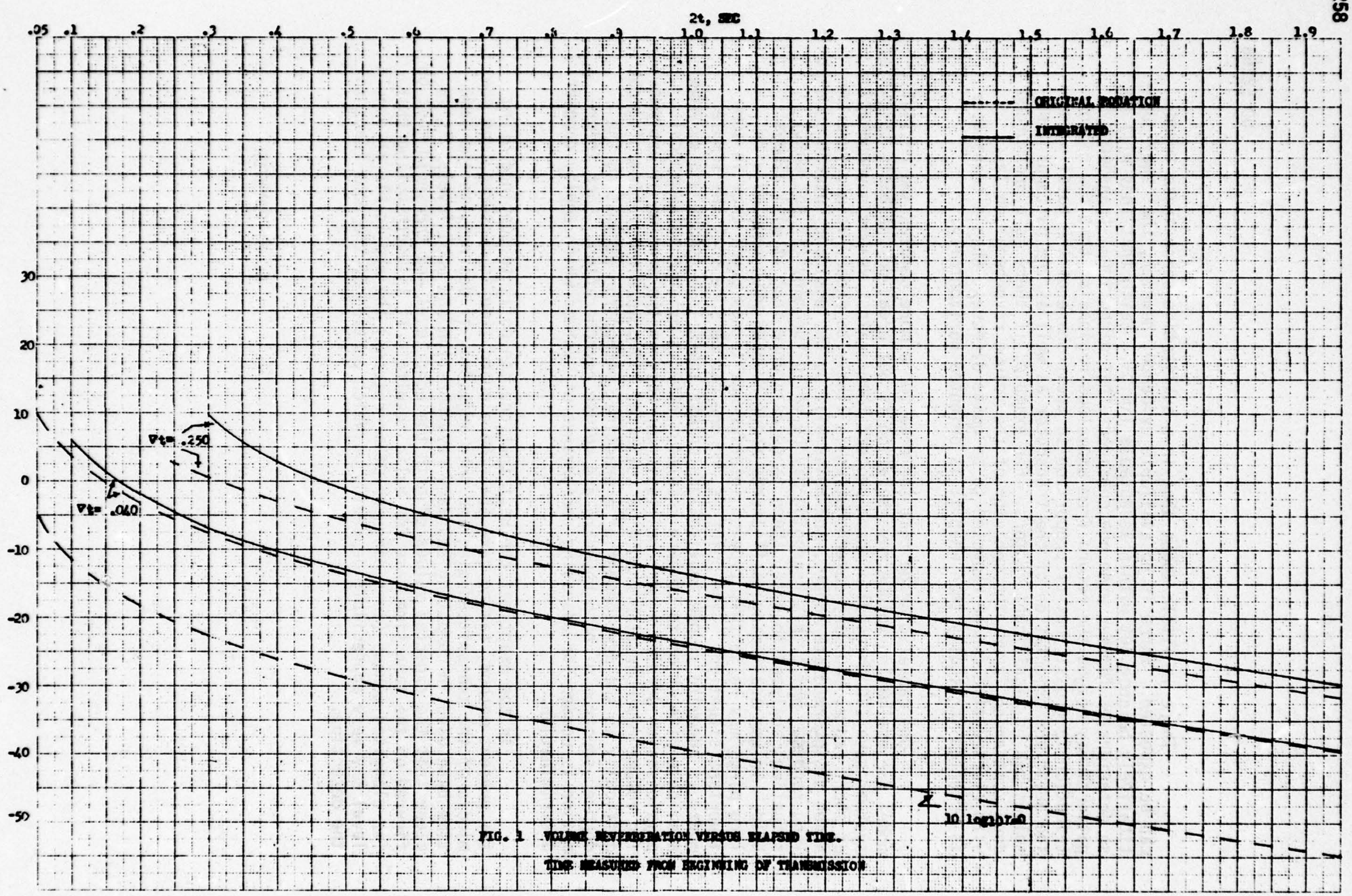


FIG. 1 VOLUME REVERBERATION VERSUS ELAPSED TIME.
TIME MEASURED FROM BEGINNING OF TRANSMISSION

APPENDIX E

MEMORANDUM

P80203/ABP:sln
24 November 1965

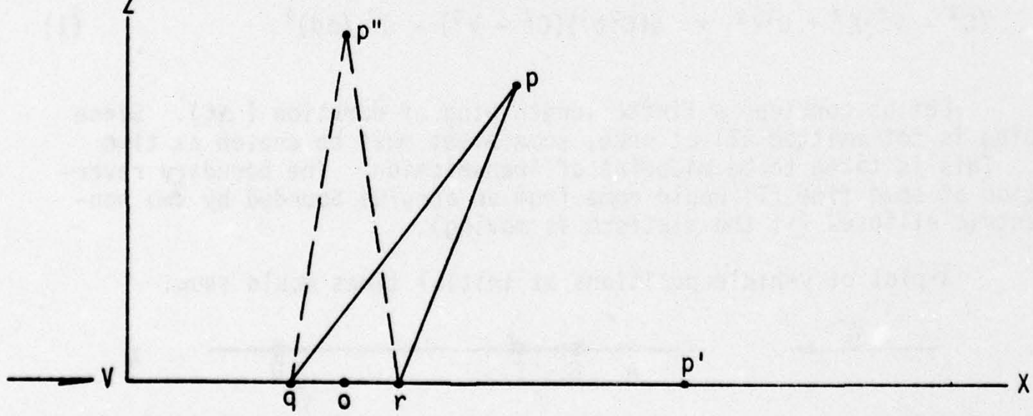
From: Code P80203
To: Code P802

Subject: Assessment of the Effect of Platform Translation on the Area Returning Boundary Reverberation at a Given Time.

Math models for computing expected levels of reverberation nearly always include the concept that the return at any given time comes from scatterers contained in a sphere-like shell. The outer and inner surfaces are determined by the sound velocity multiplied by one-way travel times (equal to one-half the elapsed time in the ping cycle) plus and minus one-fourth of the ping duration. This concept fits volume reverberation at least for times less that that required for the expanding shell to be truncated by the surface or bottom. The area returning boundary reverberation is considered to be bounded by the traces of the shell boundaries on the surface or bottom. These traces will be concentric circles if the shell is truly spherical.

The approach is clearly realistic if the transducer is stationary and the sound velocity is a function only of depth. It is desirable to have some quantitative indication of the error introduced if the transducer is mounted on a moving platform. For practical cases it can be assumed that the velocity of the platform will be small in comparison with the sound velocity.

Assume that the platform is moving at a constant velocity (V) directed horizontally along the X-axis in an oceanographic coordinate system. The X-axis is taken to be at platform depth. Since the velocity of sound in the whole ocean really varies over a range of only about 10% it is reasonable to assume, as a first approximation in determining range, that the sound velocity (C) is a constant in a limited volume of water. Consider the X-Z plane in the following working figure:



Let q be the position of the platform at transmission of a particular part of the ping and r be the platform position when return is received from a point (p) after some elapsed (t). We know that the distance traveled over qpr = tC = qp'r = qp"r. The locus of all p in this plane is an ellipse with foci at q and r. If the origin is placed half-way between q and r, the equation of the ellipse is $x^2/a^2 + z^2/b^2 = 1$.

When the point is on the X-axis (at p'), z = 0 and a = op' = Ct/2. It is also known that qo = or = Vt/2. When the point is at p", X = 0, and qp" = p"r = Ct/2. Therefore,

$$b = \sqrt{(Ct/2)^2 - (Vt/2)^2} = (t/2)\ \sqrt{C^2 - V^2}$$

The distance from either focus to the origin should be:

$$\sqrt{C^2t^2/4 - (C^2t^2/4 - V^2t^2/4)} = \sqrt{V^2t^2/4} = Vt/2$$

which checks with what was known directly from the movement of the platform.

The equation of this ellipse is $\dfrac{X^2}{C^2t^2/4} + \dfrac{Z^2}{t^2/4(C^2 - V^2)} = 1$

The eccentricity of such an ellipse is $e = \sqrt{(a^2 - b^2)/a^2} = V/C$. Now C is of the order of 5000 ft/sec while a V of 15 knots is approximately 25 ft/sec. In this case e = 25/5000 = .005. For a 45-knot V, e would be three times larger or about .015. Therefore, circular assumption is not bad.

If excursions in the Y direction are allowed, the insonified area would be the surface of an ellipsoid

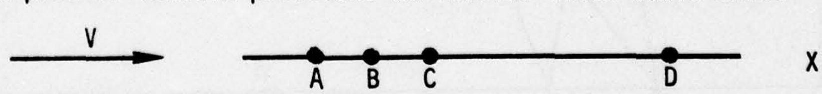
$$\frac{X^2}{C^2t^2/4} + \frac{Y^2}{(t^2/4)(C^2 - V^2)} + \frac{Z^2}{(t^2/4)(C^2 - V^2)} = 1$$

The trace of this ellipsoid intersecting a horizontal plane (e.g. surface or bottom) would be an ellipse with the same eccentricity previously computed. The equation of such an ellipse may be found by substituting (Δd), the depth differential between the platform and the boundary in question, for Z in the equation above. The new X - Y plane now coincides with the surface or bottom, and the equation for the ellipse can be expressed in the form

$$(C^2 - V^2)X^2 + C^2Y^2 = \tfrac{1}{4}(C^2t^2)(C^2 - V^2) - C^2\,(\Delta d)^2. \qquad (1)$$

Let us consider a finite length ping of duration (Δt). Since the ping is not emitted all at once, some event must be chosen as time zero. This is taken to be midpoint of transmission. The boundary reverberation at some time (T) would come from an annulus bounded by two non-concentric ellipses (if the platform is moving).

A plot of vehicle positions at initial times would show:



where A is the position at transmission of the leading edge of ping; B is the position at t = 0; C is the position at transmission of the trailing edge; and D is the position when reverberation is returned at time T. The distances A to C = (Δt)V; A to D V(T + Δt/2); and C to D = V(T - Δt/2). By alternately substituting T + Δt/2 and T - Δt/2 for the t in Eq. (1), one finds the equation for the two ellipses when the origin for each is halfway between their respective foci.

The origin of the ellipse stemming from the trailing edge of the ping will be located $V(\Delta t/2)$ further along the X-axis than the one associated with the leading edge. Choosing an origin appropriate to the midpoint of the ping and substituting in Eq. (1), one obtains the following equations:

Leading edge: $(C^2 - V^2)\left[X + V(\Delta t/4)\right]^2 + C^2Y^2 = \left[C^2(T + \Delta t/2)^2/4\right](C^2 - V^2) - C^2(\Delta d)^2$

Trailing edge: $(C^2 - V^2)\left[X - V(\Delta t/4)\right]^2 + C^2Y^2 = \left[C^2(T - t/2)^2/4\right](C^2 - V^2) - C^2(\Delta d)^2$

When y is 0 and letting subscripts L and T designate the leading and trailing edges, respectively:

$$X_L = \pm \sqrt{(C^2/4)(T + \Delta t/2)^2 - C^2(\Delta d)^2/(C^2 - V^2)} - V(\Delta t/4)$$

$$X_T = \pm \sqrt{(C^2/4)(T - \Delta t/2)^2 - C^2(\Delta d)^2(C^2 - V^2)} + V(\Delta t/4)$$

Using primes to differentiate between concepts, the similar equations for the concentric circle approximation with the origin at the same place (halfway between the midpoint of transmission and the point of reception) would be:

$$X'_L = \pm \sqrt{(C^2/4)(T + \Delta t/2)^2 - (\Delta d)^2}$$

and $$X'_T = \pm \sqrt{(C^2/4)(T - t/2)^2 - (\Delta d)^2}$$

It is clear that $X_L < X'_L$ first, by the quantity $V(\Delta t/4)$, and then by the amount the evaluation of the square root term is reduced because $(\Delta d)^2$ is multiplied by $C^2/(C^2 - V^2)$ rather than by C^2. Now $V(\Delta t/4)$ will be small in practical situations. For a platform velocity of 45 knots and at Δt of 0.25 seconds, $V(\Delta t/4) < 5$ feet. The other term is more difficult to evaluate since it depends on so many factors. At V = 45 knots, $C^2/(C^2 - V^2)$ is the order of 1.00025. However, the extent to which this factor modifies X_L depends not only on Δd, but also on Δt and the value of T for which the evaluation is desired. At the times for which the evaluation is important, the first term under the square root will be much larger than the second, so the difference in X_L and X'_L will be small.

In the forward direction the differences in X_T and X'_T (when Y = 0) is less than in the case of the leading edge since the correction for translating the origin from the center of the ellipse to the center of the circle changes sign. The two factors causing the difference tend to compensate rather than add. In the -X direction the opposite trend is seen and $-X_T$ is affected more than $-X_L$.

Another interesting case to evaluate is the one where X = 0. In this case

$$Y_L = \pm \sqrt{(C^2/4)(T + \Delta t/2)^2 - (\Delta d)^2 - (V^2/4)(T + \Delta t/2)^2 - \left[(C^2 - V^2)/C^2\right](V\,\Delta t/4)^2}$$

$$Y_T = \pm \sqrt{(C^2/4)(T + \Delta t/2)^2 - (\Delta d)^2 - (V^2/4)(T - \Delta t/2)^2 + [(C^2 - V^2)/C^2](V\,\Delta t/4)^2}$$

For the circular approximation

$$Y'_L = \pm \sqrt{(C^2/4)(T + \Delta t/2)^2 - (\Delta d)^2}$$

$$Y'_L = \pm \sqrt{(C^2/4)(T - \Delta t/2)^2 - (\Delta d)^2}$$

The difference between the corresponding Y and Y' expressions comes from the two additional terms under the square root in the equations for Y. As long as V remains very small in comparison with C, the Y values will not be materially smaller than the Y' values.

In a JP training assignment, Lee Sheldon numerically evaluated the error to be expected in a variety of cases by computing the X-axis intercepts of the ellipses and circle approximations, with the center of the circles taken as the origin. It is in this dimension that the errors are maximum. Vehicle speeds of 45 and 15 knots were assumed. Results are shown in Table 1 when the sound velocity is considered to be a constant 5,000 ft/sec and the ping duration is 0.25 sec. Three elapsed times were examined in combination with two depth differentials between vehicle and boundary. Table 2 shows some results obtained by using ray tracing data in a refracting medium.

On the basis of these data, it would appear that for vehicle speeds less than say 50 knots the circular approximation of the boundaries of the annulus returning reverberation at a given time introduces only very minor errors. These are certainly negligible in the light of other sources of error, such as in the determination of the scattering coefficients. The maximum error in X occurs to the rear of the vehicle where pattern discrimination generally is high.

Another study by Charles Williams, a summer employee, showed that, for vehicle speeds up to 50 knots, the difference between the initial path angles from the respective vehicle positions at transmission and reception of reverberation from selected points on the surface amounted to less than 1° for a wide variety of situations. The errors were largest when geometries and elapsed times were such that the effective paths were steep. Since, in computing boundary reverberation, the circular approximation uses what is essentially a median of these two angles for both transmission and reception in computing pattern losses, the circular assumption should provide an adequate model for most practical cases.

All of the above discussion assumes that the boundaries and vehicle velocity vector are horizontal. The situation is much more complicated when one or both of these conditions are not applicable. Work in this area is needed.

A. B. POYNTER

TABLE I

C = 5000 ft/sec; Δt = 0.25 sec; Errors Computed with Ellipse as Standard

PARAMETERS			CIRCLES		ELLIPSES							
			± X-AXIS		+ X-AXIS				- X-AXIS			
							% ERROR				% ERROR	
y ft/sec	d ft.	T sec.	X_L yd.	X_T yd.	X_L yd.	X_T yd.	X_L	X_T	X_L yd.	X_T yd.	X_L	X_T
25.33	250	0.250	301.18	62.50	300.53	63.02	+0.18	-0.83	301.68	61.97	-0.35	+1.69
		1.000	933.79	724.39	933.18	724.85	+0.07	-0.06	934.23	723.79	-0.11	+0.15
(15 kt)		10.000	8437.09	8228.74	8435.83	8228.56	+0.01	0.01	8436.88	8227.50	0.01	+0.01
	3000	1.425	817.56	416.67	816.94	417.12	+0.08	-0.11	817.99	416.07	-0.05	+0.25
		2.625	2061.97	1827.64	2061.25	1827.99	+0.03	-0.02	2062.30	1826.94	-0.02	+0.04
		10.000	8378.03	8168.18	8376.78	8167.99	+0.01	0.01	8377.83	8166.94	0.01	+0.02
76.0	250	0.250	301.18	62.50	299.53	64.01	+0.55	-2.36	302.70	60.89	-0.50	+2.64
		1.000	933.79	724.39	931.99	725.80	+0.19	-0.19	935.16	722.64	-0.15	+0.24
(45 kt)		10.000	8437.09	8228.74	8433.55	8228.42	+0.04	0.01	8436.72	8225.26	0.01	+0.04
	3000	1.425	817.56	416.67	815.65	417.88	+0.23	-0.29	818.81	414.71	-0.15	+0.47
		2.625	2061.97	1827.64	2059.86	1828.74	+0.10	-0.12	2063.02	1825.57	-0.05	+0.11
		10.000	8378.03	8168.18	8374.49	8167.86	+0.04	0.01	8377.66	8164.69	0.01	+0.04

TABLE II

Refracting Medium; Δt = 0.25 sec; Errors Computed with Ellipse as Standard

76.0	3000	1.875	1307.0	1035.0	1305.0	1035.6	+0.15	-0.06	1308.1	1032.4	-0.08	+0.06
(45 kt)		2.625	2032.0	1799.0	2030.0	1800.1	+0.10	-0.06	2033.1	1796.9	-0.05	+0.12
		11.000	9118.5	8911.0	9116.9	8911.8	+0.02	.01	9120.1	8908.7	-0.02	+0.03

APPENDIX F

MEMORANDUM

P3502/ABP:dd
Reg No. P3502-136
15 April 1968

From: Code P35021
To: Code P3502

Subject: Comparison of Computer Models for Obtaining the Expected Level of Boundary Reverberation as a Function of Elapsed Time.

References:

A. NAVORD Conf Report 5606, "Analytical Methods for Predicting the Acoustic Performance of Homing Torpedoes in Circular Search" (NOTS 1818) of 26 July 1957.

B. NOTS P80203 memo, "Review of Mathematical Model for Computing Boundary Reverberation", of 20 January 1965.

C. NOTS P80203 memo, "Error from Using 10 $\log_{10}\tau$ as a measure of Effective Train Length when Computing Boundary Reverberation", of 18 January 1965.

D. NUWC P35021 memo, "Integration of Power under a Curve Plotted in Decibels", of 10 January 1968.

E. NAVORD Report 4962, "A Study of the Effects of Refraction on Reverberation (NOTS 1284) of 7 November 1955.

F. NOTS P80203 memo "Doppler Content of Boundary Reverberation Due to Vehicle Translation-Refractive Environment Case" of 21 December 1965.

Over the last few years various aspects of the problem of computing the expected level of boundary reverberation in a refractive medium have been re-examined. The purpose of this memorandum is to compare the results obtained from four models selected as being practical for computer application and to assess their relative accuracy. In general, the accuracy to be anticipated increases with the complexity. The study is made in terms of surface reverberation. In the case of a flat bottom, bottom reverberation is computed in an identical fashion.

Model I has been used since the inception of our first ray-tracing program and was based on Reference A. The form of the equation used in the program is

$$R_s = S - 2H + 10 \log_{10} m_s - J_s + 10 \log_{10}\tau + 10 \log_{10} \frac{X}{\cos \theta_0}$$

where

- R_s is the expected level of surface reverberation at a given time
- S is the on-axis source level in dB re one microbar at one yard
- 2H is the two-way transmission loss along the ray path

- $10 \log_{10} m_s$ is the surface scattering coefficient in dB as a function of grazing angle of the ray path at the surface
- J_s is the boundary reverberation index of the transducer
- $10 \log_{10} \tau$ is the effective annulus width in dB
- $10 \log_{10} \dfrac{X}{\cos \theta_0}$ allows for the fact that horizontal spreading is compensated for in the outgoing direction by the fact that the whole annulus back-scatters toward the source. Here X is the horizontal range traversed by the ray to boundary intercept, and θ_0 is the ray angle at the source.

In this model $10 \log_{10} \tau = \frac{V(\Delta t)}{2}$ where V is the nominal velocity of sound (1667 yd./sec.) and Δt is the ping duration in seconds.

It was shown in Reference B that the model produces the best results if the elapsed time T is considered to be measured from the midpoint of transmission so that, in effect, T = 2t, the two-way travel time of the ray being run. This ray then reaches the surface near the mid-range of the insonified annulus so that the associated value of transmission loss, J_s, and grazing angle are reasonable representative for the annulus as a whole.

Reference C showed that τ was not a good measure of annulus width where steep paths are involved. Model II is a simple attempt to improve this situation. The only change from Model I is to compute

$$\tau = \frac{V_s}{2} \frac{t}{\cos \theta_s}$$

where V_s is the velocity at the surface and θ_s is the grazing angle of the path at the surface. This is largely an intuitive "fix", and it is being tried here for the first time. Obviously, both models I and II get into trouble when $T = 2t_{\theta_c} < 2t_{-90°} + \Delta t/2$ because the full ping is not insonifying the surface. In Model II a test was made to ascertain whether or not this criterion was being met. If not, it was assumed that the paths would be sufficiently steep to approximate straight lines. Since τ would equal the horizontal range to surface intercept of the leading edge,

$$\tau \approx \frac{\overline{V}}{2} \left(2t_{\theta_0} + \frac{\Delta t}{2}\right) \cos \theta_0 .$$

$\overline{V}$ is the average velocity over these straight paths and can be found by dividing the source depth by the one-way travel time of the -90° ray.

As pointed out in Reference B, the above models in some situations could be expected to give inaccurate results for long pings even if the annulus width were modeled adequately. Values of transmission loss, vertical pattern losses, and grazing angle found by tracing a ray to any single point in a wide annulus cannot be expected to represent those over the full annulus, particularly in regions where one or more of the parameters is varying rapidly. Moreover, at elapsed times sufficiently short so that all of the ping has not insonified the surface, the ray path for which the two-way

travel time is computed no longer reaches the surface near the mid-point of the annulus. These difficulties are overcome in the next model. Model III in this study follows the scheme delineated in Reference B and computes boundary reverberation levels as in Model I with 10 log set to zero. This is equivalent to a one-yard annulus width. From the data produced by the ray-tracing program, both the two-way travel time (2t) and computed reverberation level (R_0) are plotted as functions of the horizontal range (x) to surface intercept as determined for each ray path. For each desired elapsed time (T), which can be selected at will, the limits of the true annulus corresponding to Δt is found by evaluating x on the time curve for $T - \frac{\Delta t}{2}$ and $T + \frac{\Delta t}{2}$.

The actual reverberation level at time T is then found by summing in random phase the contributions of each one-yard increment between the above limits as shown by the R curve. The summation was a hand operation in Reference B but it was computerized in this study in the manner delineated in Reference D.

These first three models all require a table input to account for the weighting effect of the horizontal patterns on the reverberation return. These data are combined with vertical pattern losses and a geometric correction for transducer pitch to give J_s the boundary reverberation index of the transducer. NUWC computer program 819001 is available for generating the data for the table showing the average weighting effects of the horizontal patterns. For a non-turning transducer, this is a single value. The equations for evaluation J_s were developed in Reference E. Because of certain simplifications introduced in the concept, the accuracy of the results is suspect when the transducer axis is tilted considerably and/or when the steep sound paths are producing the reverberation at time T. The basic simplifications are the assumptions that pattern effects can be obtained when only the patterns in the cardinal planes are known and that the variables can be separated for integration.

Model IV uses a different approach. The concept was developed in Reference F for the purpose of computing the boundary reverberation returned in doppler bands when the doppler is introduced by own vehicle speed. Briefly, the insonified annulus is divided into incremental areas bounded on two sides by equal doppler lines and on the other two sides by equal travel-time circles. Rays are traced to the corners of these little areas yielding the coordinates of the corners in space so that the areas of each can be computed. Other ray data permit evaluation of the transmission loss, scattering strength, and the transmit and receive pattern losses needed to ascertain the back-scatter received from each incremental area. When these contributions are summed in random phase (expressed in intensities) for each doppler band, one gets the reverberation levels as might be observed by a system which processes signals in narrow frequency bands. The summation of contributions from all bands gives the total surface reverberation received. (The first three models yield only the total reverberation as might be observed in

... have an average ... Later this was changed to 35 dB down as noted in Figure 1. These data are useful in interpreting the reverberation levels which will be shown later. Transducer pitch angles of 0° and 10° up were used. The same table of scattering coefficients versus grazing angle were used in Models I through III. These values were each reduced by $10\log_{10}2\pi$ for Model IV so that they would reflect scattering strength as required in that model. Computations were made for all four models using each of two quite different sound-velocity profiles in order to see that the conclusions arrived at from comparing the results obtained from the four models were not prejudiced by some over-riding characteristic of the profile. The velocity profiles are shown in Figure 2. A transmit frequency of 20 kHz was chosen, and an appropriate table giving the attenuation losses as a function of depth was input with each profile. Ping durations of 40 and 250 milliseconds were considered. Model IV assumed a horizontally-directed, straight-running platform at a 40-kt. speed, and reverberation was computed in 1/4-kt. doppler bands. The vehicle speed is an artifax here since we are interested only in the total return. It was selected to insure sufficiently small area increments to yield good accuracy. Actually, the results can apply to a stationary system since speed is not considered in the other three models. A 120-dB source level was used in all cases. The transducer is assumed to be at a 1000-ft. depth.

It proved to be impracticable to show the results for all four models on a single graph because of the clutter. Consequently, for each combination of environment, ping length, and pitch angle, two figures will be used. The first will compare the data from Models I, II, and III. On the second graph the data from Model III will be repeated and compared with that from Model IV. This is a natural grouping for our purpose in that the first three models assume common values for J_s. Then by comparing the most accurate of these models with the doppler-band method, one should be able to obtain an idea of the accuracy with which we now evaluate J_s.

Figure 3 shows the surface reverberation levels computed in Environment A for a 40-ms ping and a 0° pitch angle. With the source at a 1000-ft. depth the middle of the ping has a two-way travel time to the surface of about 0.405 seconds via a vertical path ($\theta_0 = -90°$). The velocity profile is dominated by negative gradients and the -15° ray is very nearly the last to reach the surface. The 3.8-sec. ping interval was selected with this in mind. The combined vertical patterns largely determine the peaks and valleys in the respective reverberation curves. Of course, the trailing off of the reverberation at times greater than about 1.7 seconds is caused by the transmission loss increasing more rapidly than the pattern losses are decreasing. In addition, the scattering coefficient tends to be smaller at the lower grazing angles. All three models tend to be in good agreement beyond about 1.05 seconds when the major lobes of the vertical patterns govern. The integrated method (Model III) tends to smooth out the narrow peaks and valleys generated by the other two models. It is undoubtedly the more accurate since it does not consider the particular values of pattern and transmission loss associated with a data point as being necessarily representative of the entire annulus returning reverberation at that elapsed time. The cosine correction to the annulus width used in Model II seems to yield good agreement with the integrated model over the broad minor lobe. (The surface velocity used in Model II is not significantly different than the nominal velocity used in the first model). However, it tends to drastically over-compensate on the other minor lobe as can be expected; the cosine is rapidly approaching zero as the path angle approaches -90°.

The results for the 250-ms ping under otherwise similar conditions are presented in Figure 4. The curves for Models I and II have the same shape as in Figure 3; they are merely at a higher level because of the longer ping. The agreement between the three models is still good beyond about 1.2 seconds, but the importance of the integrated method when long pings are used is quite evident at shorter elapsed times. The results of its use are most striking in modeling the onset of boundary reverberation. The flat portion of Model III curve centered at about T = 0.35 is due to the narrow lobe being fully covered by some portion of the effective train length while the rest is contributing negligible return. With the quarter-second ping the broad minor lobe dominates the return until such time that the contribution from the major lobe begins to build up.

Figure 5 shows the corresponding data for the 40 ms-ping in Environment A when the transducer is pitched up 10°. The gross effect is to make the peaks and valleys in the expected level of reverberation more narrow than was the case when the transducer was directed horizontally. This is to be expected since each path making a particular angle with the transducer axis is 10° steeper in the oceanographic coordinate system and returns reverberation at a shorter elapsed time. The difference in results for the three models are of the same order as were found for the horizontally-directed transducer.

Figure 6 shows the results under the same conditions as in the preceding paragraph except that the ping duration is increased to 250 ms. As anticipated, the integration method (Model III) gives a decidedly different curve than the other two models at times less than one second. This model permits development of the onset of the surface reverberation starting where

$$T > 2t_{90°} - \frac{\Delta t}{2}$$

The flat portion starting at 0.3 seconds covers the time region where the narrow minor lobe alone is contributing a substantial return. The level then rises as the next lobe also contributes. The drop-off starting at about 0.52 seconds is caused by the narrow minor lobe dropping out of the picture. The next flat portion occurs when only the second lobe is contributing materially. The shape of the remaining portion of the curve is rather obvious.

The next group of figures (Figure 7-10, inclusive) compare the results from the same three models when Environment B applies. Note that the dB scale has been changed. The two-way travel time from 1000 feet to the surface via a vertical path is about 0.418 seconds. Surface reverberation is continuous thereafter to an elapsed time greater than the ping interval which was taken as 10 seconds. The path yielding this two-way travel time to the surface has an initial path angle of approximately +3.7° with respect to the horizontal. At the scale plotted there is no discernible difference in the results from Models I, II, and III at elapsed times beyond 1.6 seconds. Therefore, only the first portion of the ping interval is shown in these figures.

For Environment B the coincidence of the three curves begins shortly after the major lobe of the patterns assume dominance. This occurs at an earlier time than in Environment A for a given pitch angle since refraction is such that a ray with a given initial angle will tend to reach the surface much sooner. Also, in the case of Environment B, the reverberation declines at a lower rate after the peak is reached so that the integrated curve does not begin to rise above the levels shown for the other two models later in the ping cycle as occurred to some degree in the case of the first environment.

At the shorter elapsed times, the differences in the three curves tend to be quite similar (for each of the four combinations of ping length and pitch angle) to those prevailing for the same combination when Environment A applied. Model III, the integration method, tends to smooth out the peaks and valleys as shown for the other two models, particularly when they are narrow with respect to the ping duration. This method also is better in modeling the onset of boundary reverberation. For the shorter ping, the annulus width correction applied in Model II appears to compensate adequately in the region of the broader minor lobe, but it tends to overcompensate when the paths become steep enough for Cos θ_0 to become very small. The introduction of a 10° up pitch produces about the same effect in both environments.

Now we turn to the interesting task of comparing the results of the integrated method with those of the doppler-band model. Figure 11 shows the data obtained in Environment A for a zero pitch angle and a pulse length of 40 ms. The larger scale again is used for clarity. Earlier, the hypothesis was advanced that agreement between data from the two models at the longer elapsed times (where flat paths obtain) would be a good indication that the doppler-band program (Model IV) was functioning as intended. The agreement is reasonably close over the region dominated by the major lobes of the patterns, but there remains a slight difference at 3.8 seconds. However, in this environment the path angles yielding this elapsed time is still over 15° off the transducer axis. As anticipated, some rather substantial differences are observed in the region dominated by the minor lobes of the patterns. Since there is little reason to suspect that Model IV is materially less accurate in this region than in any other, the evidence tends to substantiate fears that the current method for evaluating J_s is in error when large off-axis path-angles are involved. In this example (as in subsequent ones where the transducer is oriented horizontally) the differences in reverberation levels at the peaks are rather moderate (no more than 2 dB). The wide troughs are something else again. The higher minimums observed in Model IV results are believed due to the complexity of the minor lobe structure. For the results shown in Figure 11, all of the incremental areas contributing a return at an elapsed time of about one second are not subjected to maximum pattern loss. Of course, in practical applications the actual values in the valleys are not likely to be important since the interference level probably will be dominated by volume reverberation or noise at corresponding times.

Figure 12 shows comparable data when the ping duration is increased to 250 ms. The reverberation levels agree to about the same degree as for the shorter pulse over the peaks, and the differences in the valleys have been reduced. The narrow peaks and the valleys have been smoothed considerably.

Figures 13 and 14 compare the results from Models III and IV for ping lengths of 40 and 250 ms, respectively, when the transducer is pitched 10° up. Environment A still applies. The extent of the agreement between the respective curves is substantially the same as for the comparable cases when zero pitch was used except at the shortest times shown. It is felt that when path angles are involved which approach-90° the geometric correction factor,

$$-10 \log_{10} [\text{Cos } (\theta_0 - \xi)/\text{Cos } \theta_0]$$

(where ξ is the transducer pitch angle)

in J_s over-compensates and makes the levels a few dB higher than they should be for Model III. At the longest times (where the initial path angles are approaching -15°) the angles with respect to the transducer axis are approaching -5°. One might expect the two curves to coincide. The one for the integrated method is still a few tenths of a dB above the other one as it was for the zero pitch cases. This leads one to suspect that the Cos $(\theta_0 - \xi)$ / Cos θ_0 correction is slightly overcompensating even at these angles.

Figures 15 through 18 compare data from Model III and IV for the various combinations of pulse lengths and pitch angles when Environment B applies. In all four figures the two curves come into coincidence near or before an elapsed time of about 1.9 seconds which, in Environment B, corresponds to an ititial path angle of approximately -12 degrees. For zero pitch, Figure 15 and 16, the agreement persists to the 10-second ping interval which corresponds to a path angle of approximately +4 degrees. This is the most convincing evidence that Model IV is properly programmed since it is under these conditions that J_s should be most accurate.

For pitch angles of 10 degrees up, Figures 17 and 18, the near perfect coincidence begins near 1.7-sec. elapsed time, corresponding to a path angle of about -13.5 degrees. Note, however, that this corresponds to -3.5 degrees from the transducer axis. On the basis of path angle alone one might think that coincidence should occur earlier. It is suspected that the delay is caused by the $\cos(\theta_0 - \xi)/\cos\theta_0$ term in J_s over-compensating for the pitch. The agreement persists to about an elapsed time of 7 seconds, at which time the reverberation level computed by means of the integration method begins to fall slightly below that computed by Model IV. At 7 seconds the initial path angle is about +0.6 degrees while the angle with respect to the transducer axis is approximately +10.6 degrees. It seems possible that the slight divergence from 7 to 10 seconds results from the $\cos(\theta_0 - \xi)/\cos\theta_0$ correction to J_s tending to under-compensate at appreciable off-axis angles when the sign of the pitch angle is opposite that of the path angle. However, the evidence is by no means conclusive since the divergence does not increase consistently as the elapsed time approaches 10 seconds. The differences are so slight that they well may be due to errors in the numerical integration process in one or both models. Although data points are taken at relatively short intervals, they are still at finite intervals apart, and linear interpolation is used.

For the relatively short elapsed times (less than 1.5 sec.), the differences in results from the two models in Environment B are essentially the same as they were when Environment A was used with one exception. For both ping lengths with zero pitch, the minor lobe at minimum time for the integrated method (Model III) is a little over one dB lower than for the doppler method for Environment B computations. For Environment A the difference is about the same amount but in the opposite direction. This is attributed to the fact that we changed the constant pattern value assumed for the back portion of the pattern as was announced earlier. The transition was most severe when Environment A was used, and these results are considered to be unreliable. For the 10 degree up pitch, this portion of the vertical pattern does not come into play in Model III.

It was decided to make one further effort to clarify the effect of transducer pitch by computing a case involving a 40 degree up attitude. As other conditions it was decided that Environment B and a 40-ms ping would combine to produce the most useful results. Remaining parameters have the same values used throughout the study. Figure 19 compares the results produced by means of Models III and IV. The on-axis ray ($\theta_0 = -40°$) reaches the surface at T = .647593 sec. At this pitch angle, at least in Model III, the major lobe of the pattern controls the reverberation

level over a relatively small portion of the total ping interval. The first minor lobe above the transducer axis controls the level of the peak following the onset of surface reverberation. The first minor lobe below the axis governs the reverberation level over the last 75% of the 10-sec. ping interval. It is difficult to visualize physically just how the patterns interact in Model IV.

In comparing the results from the two models, it is noted first that the two curves coincide only briefly at a time centered about T = 0.85 sec. This corresponds to an initial path angle in the vertical plane of -29.1 degrees or +10.9 degrees with respect to the transducer axis. At times shorter than this the curves diverge with the integrated method giving the higher values. At about 0.43 sec. the separation is nearly 8 dB. At times greater than 0.85 seconds, the curves cross and there again is an increasing difference but with the integrated method yielding the lower values. The large difference in the vicinity of 1.9 seconds can be attributed to the fact that the null in the vertical plane patterns is not characteristic of the whole annulus which returns reverberation at times of this order. If this region is discounted, then the remainder of the ping cycle shows a very gradual increase in the difference between the two curves from about 1.7 dB at 2.5 sec. to 2.8 dB at 10 sec.

The above data would seem to support certain conclusions. In the first place, the doppler-band method (Model IV) is working properly. Although it is judged to be the most accurate way of computing the expected level of boundary reverberation, it increases the computing time by a factor of ten over that required by any of the other methods. Therefore, its general use as part of the ray tracing program is not recommended. It should be a separate program reserved for special cases. Even then it requires insertion of appropriate equations for the transducer patterns for obtaining pattern losses in any directions. Alternately, one could use a matrix of measured patterns in such a large number of planes that accurate results could be obtained by interpolation.

The other three models have in common the inaccuracy resulting from errors in computing values of the boundary reverberation index, for various geometries which obtain, as functions of transducer depth and elapsed time in the ping cycle. The evidence is that the errors inherent in the present method of computing J_s are small for path angles falling in the major lobe of the transducer when the pitch angle is not much greater than 10 degrees from the horizontal. For small pitch angles, the error in the first minor lobe may be tolerable in veiw of the usual uncertainty as to the proper values to assign for the scattering coefficient per unit area. Paths falling in the nulls between pattern lobes lead to values for J_s which are too high, but in general the boundary reverberation at corresponding times will be below other types of interference and is of no practical consequence. Model I, the method presently incorporated in our ray tracing program, also suffers from underestimating the width of the insonified annulus as the paths become steeper. Model II will improve this situation over a considerable range of path angles, but this improvement is unimportant relative to another source of error common to both models I and II. Here we refer to the assumption that values of J_s and transmission loss computed to a single point in the annulus are characteristic of the entire insonified area.

Except for very short pings, this assumption can introduce substantial errors when the rate of change of one or more of these parameters is large. On this basis it seems logical to adopt the integration method (Model III) as the regular routine in the ray tracing programs. Surprisingly enough, this can be accomplished without a significant increase in run time on the computer. With a suitable selection of rays, accurate modeling of the expected level of boundary reverberation should be achieved within the accuracy inherent in J_s. If an improved method of evaluating this factor is found, the benefits will automatically accrue in the reverberation computation.

Because of the other variables involved, it is difficult to pin down the actual errors in J_s by comparing the reverberation levels computed by means of Models III and IV. During the course of this study, a technique was envisioned whereby J_s could be investigated directly.

A. B. POYNTER

OD 52258

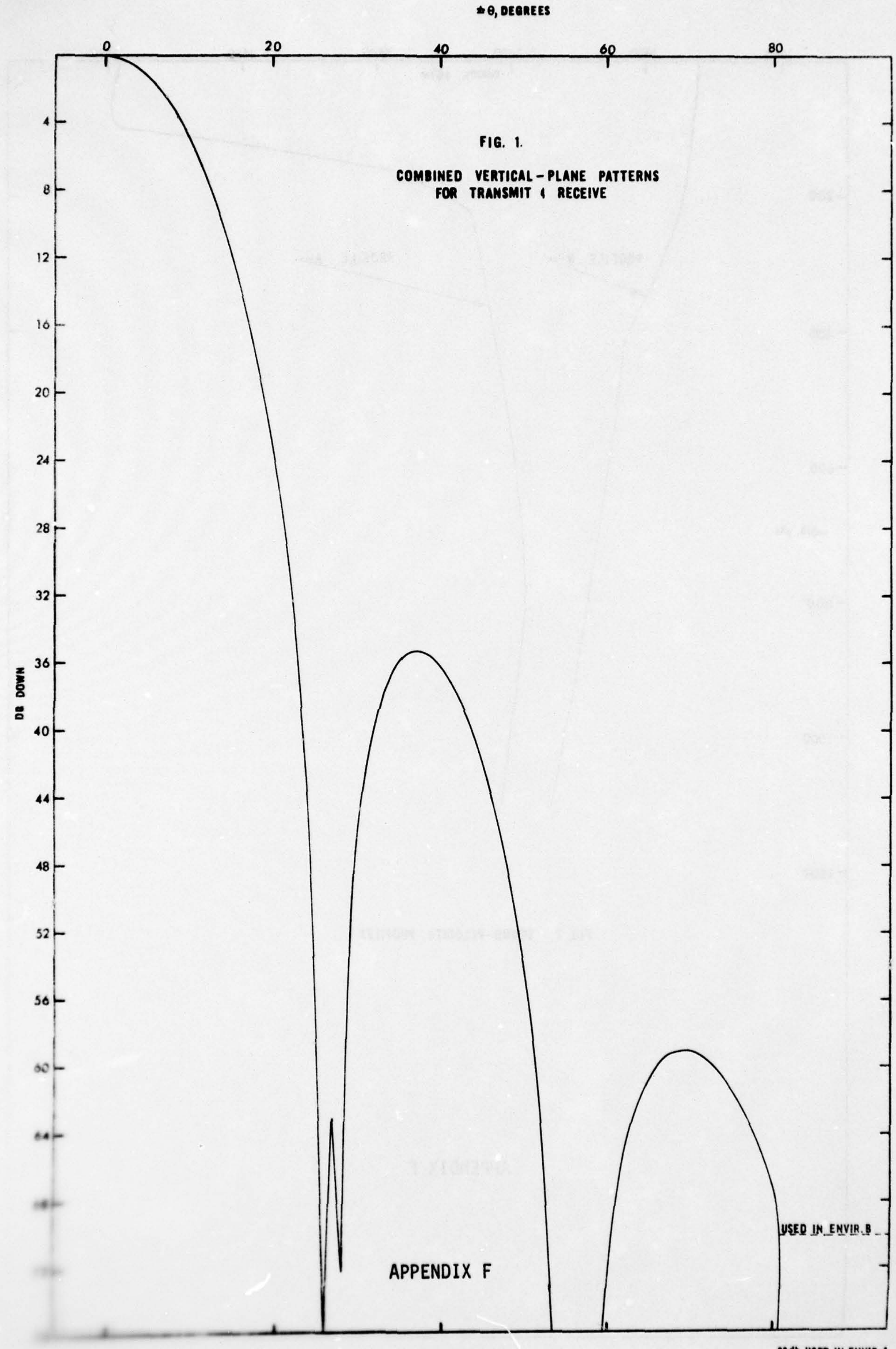


FIG. 1.

COMBINED VERTICAL-PLANE PATTERNS FOR TRANSMIT & RECEIVE

APPENDIX F

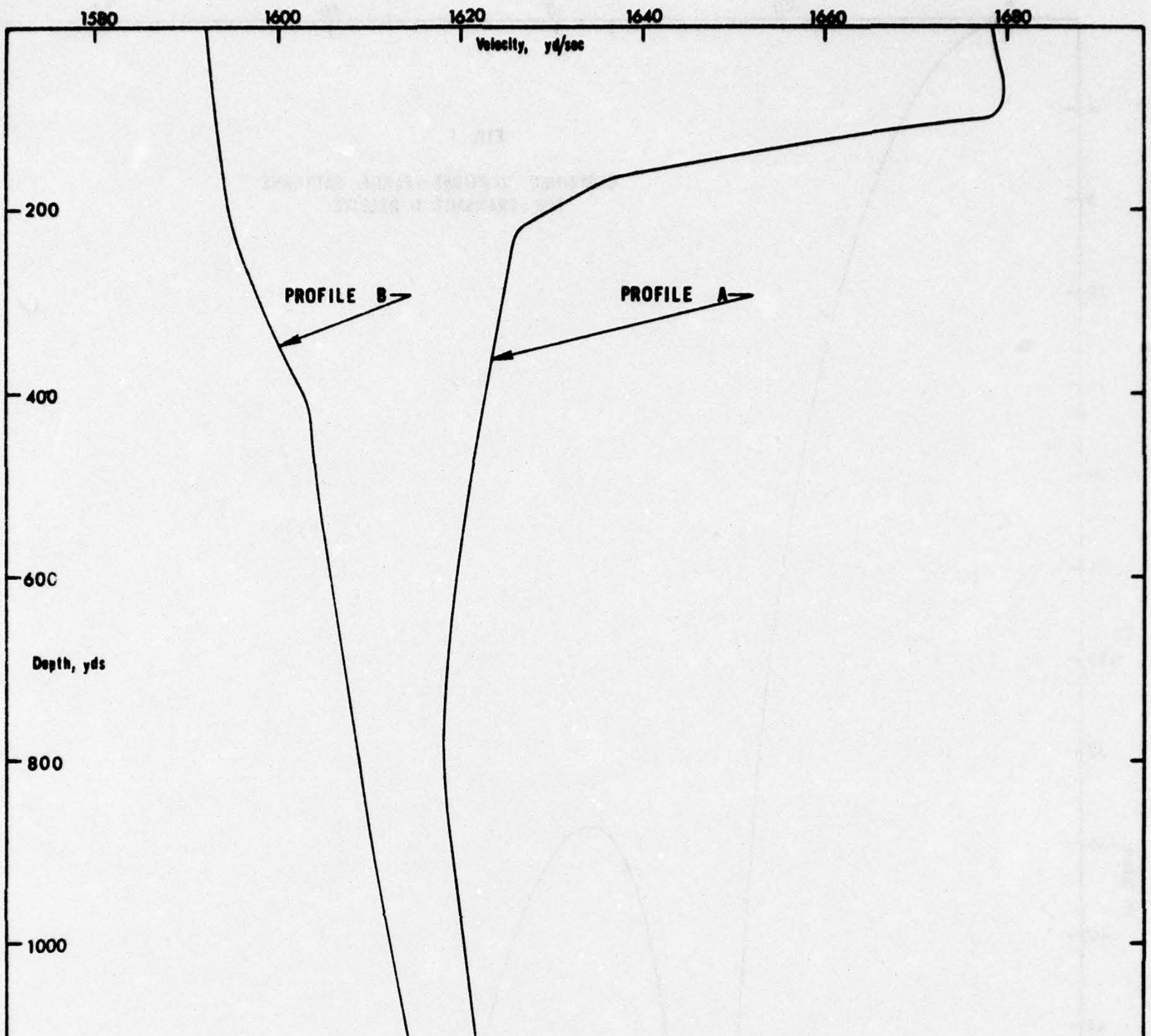


FIG. 2 SOUND-VELOCITY PROFILES

APPENDIX F

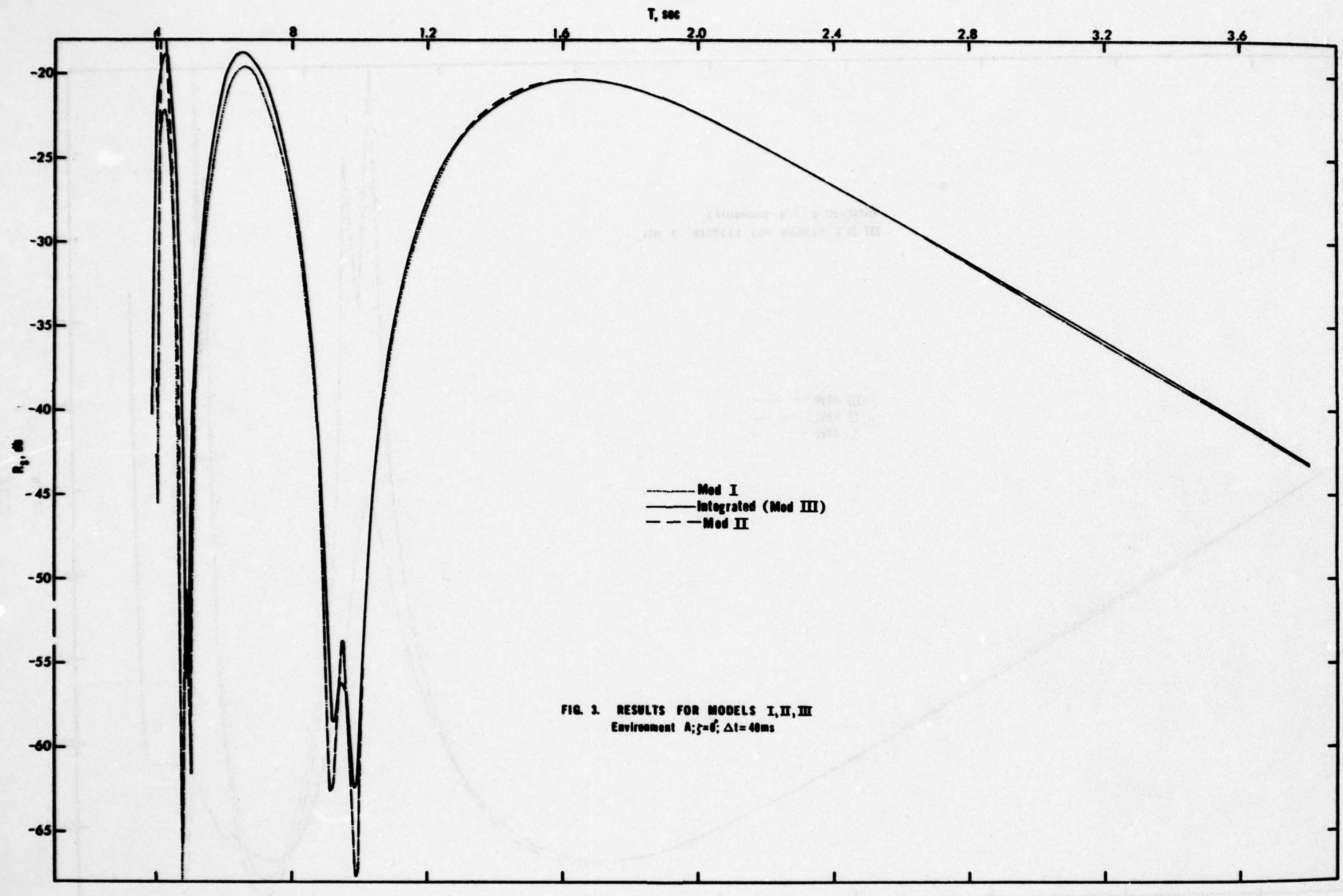


FIG. 3. RESULTS FOR MODELS I, II, III
Environment A; ζ=0°; Δt=40ms

APPENDIX F

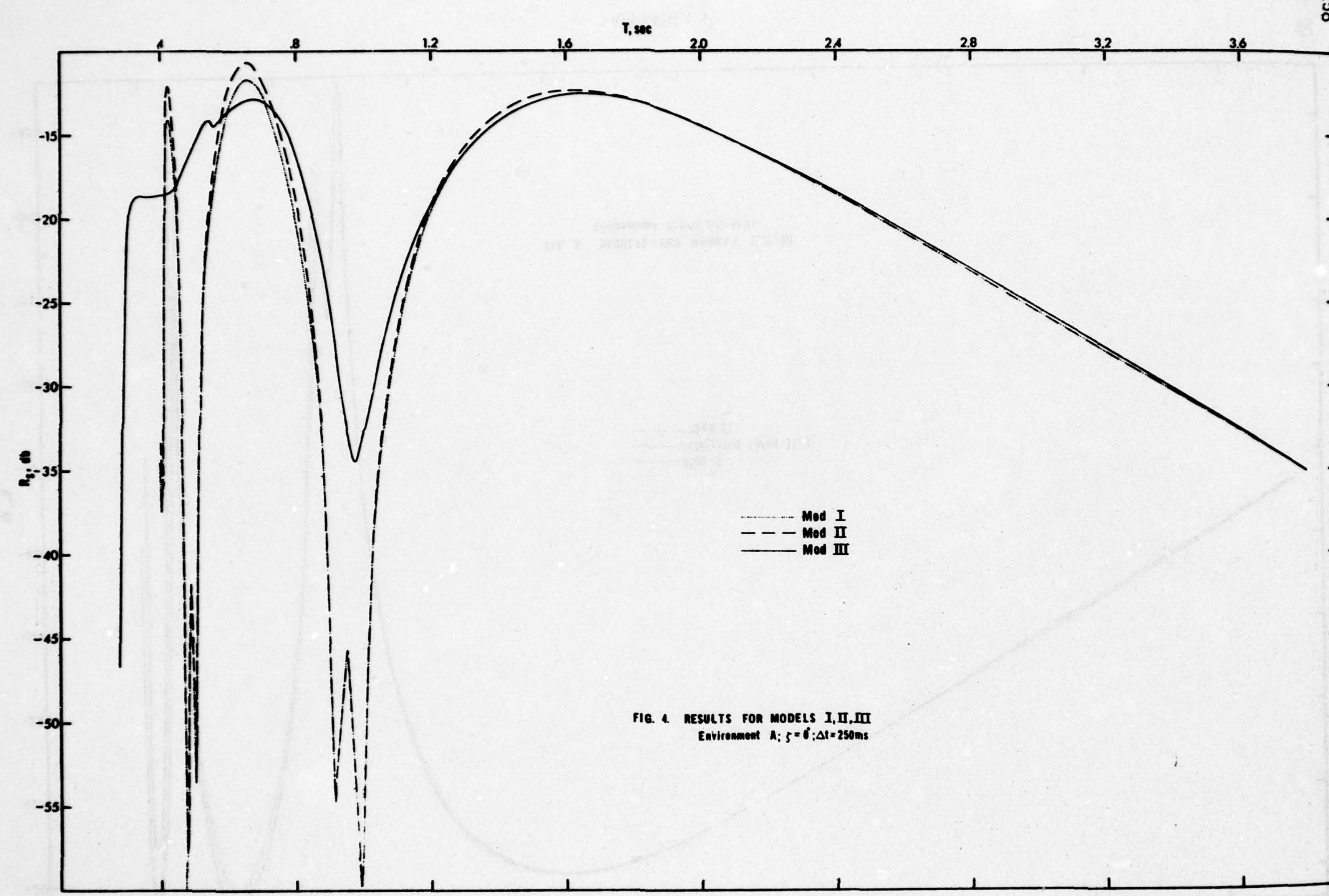


FIG. 4. RESULTS FOR MODELS I, II, III
Environment A; $\zeta = 0°$; $\Delta t = 250ms$

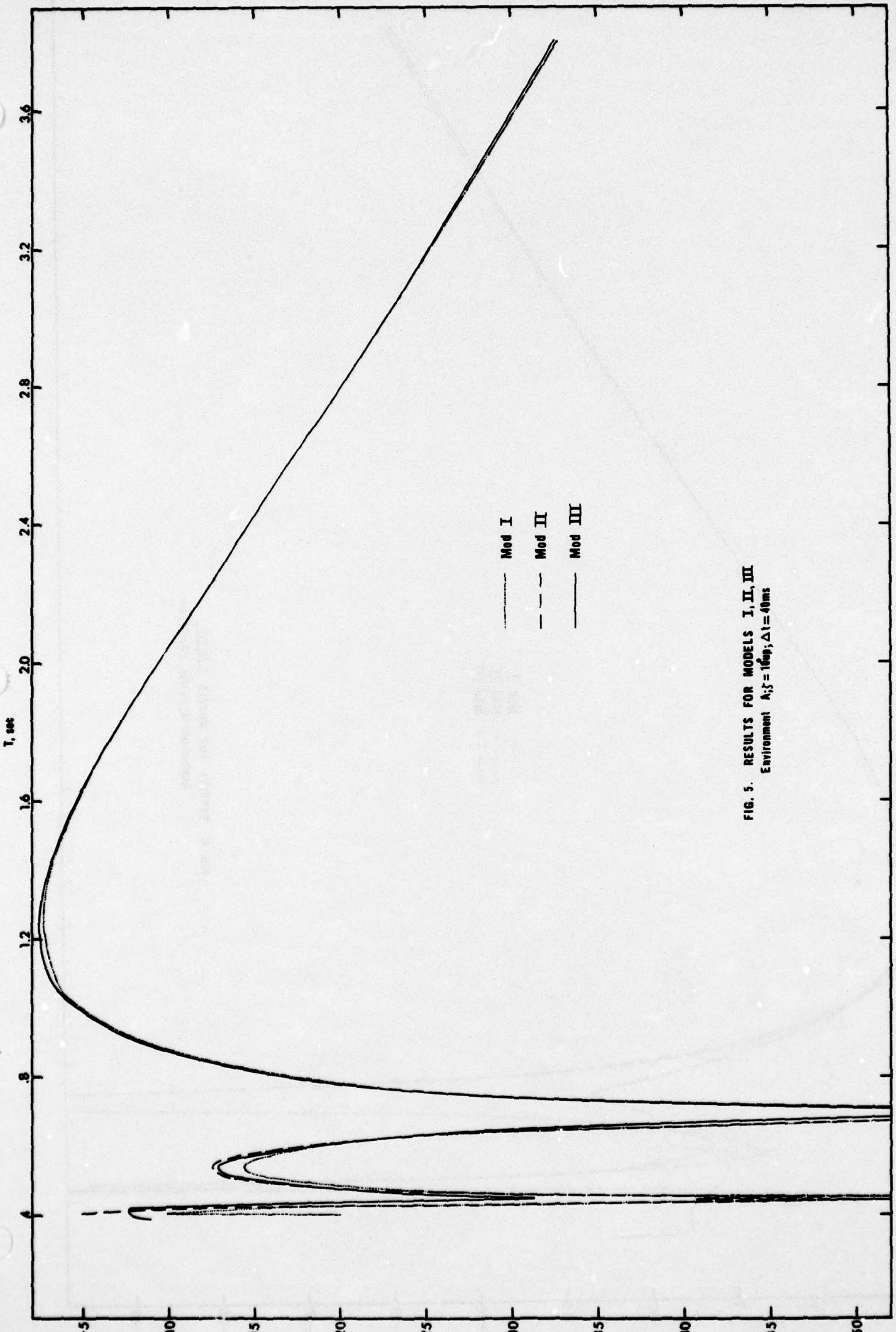


FIG. 5. RESULTS FOR MODELS I, II, III
Environment A; ζ = 10m; Δt = 40ms

APPENDIX F

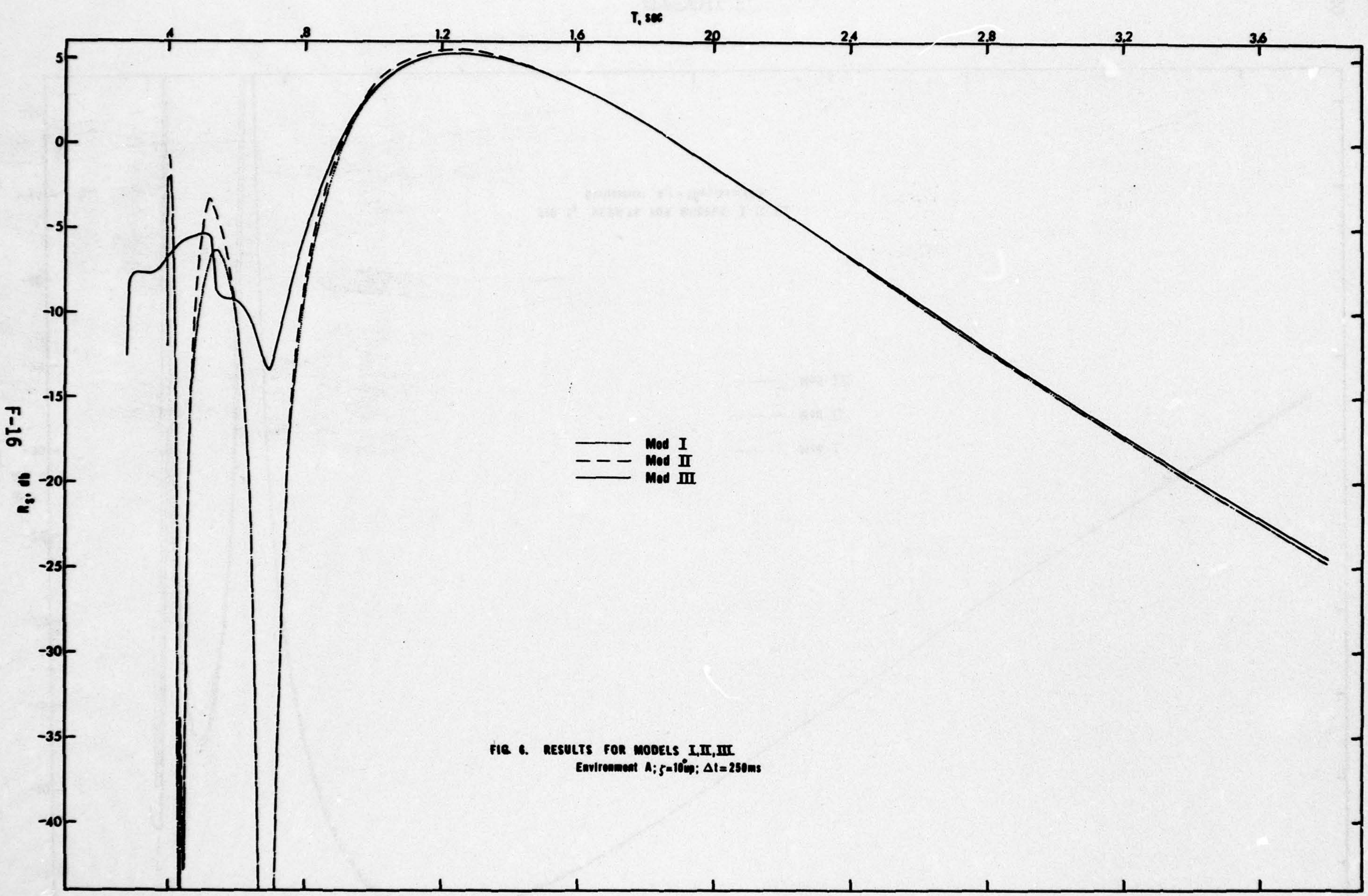


FIG. 6. RESULTS FOR MODELS I,II,III
Environment A; ζ=10°up; Δt=250ms

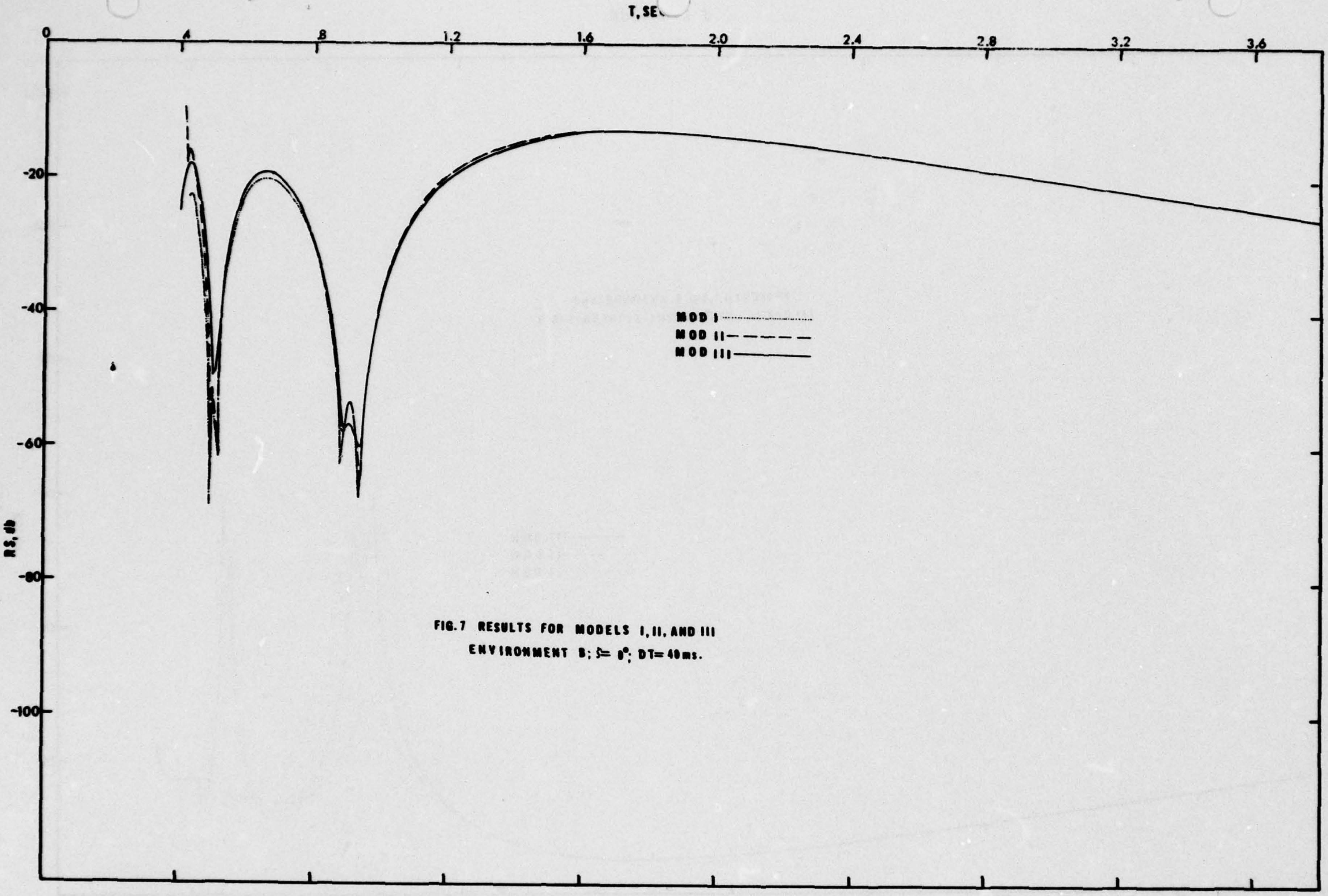


FIG. 7 RESULTS FOR MODELS I, II, AND III
ENVIRONMENT B; $\zeta = 0°$; DT = 40ms.

APPENDIX F

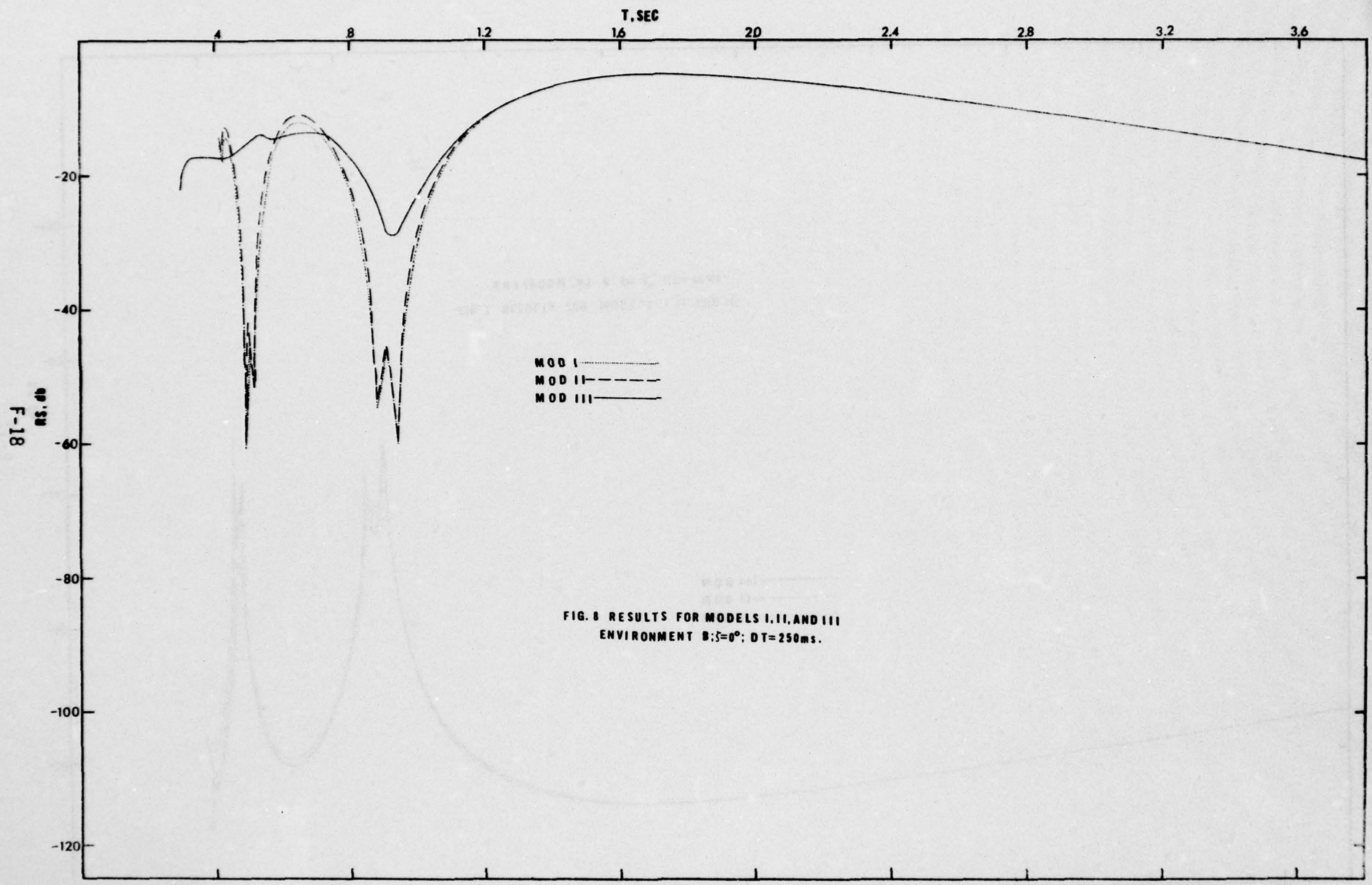


FIG. 8 RESULTS FOR MODELS I, II, AND III
ENVIRONMENT B; ζ=0°; DT=250ms.

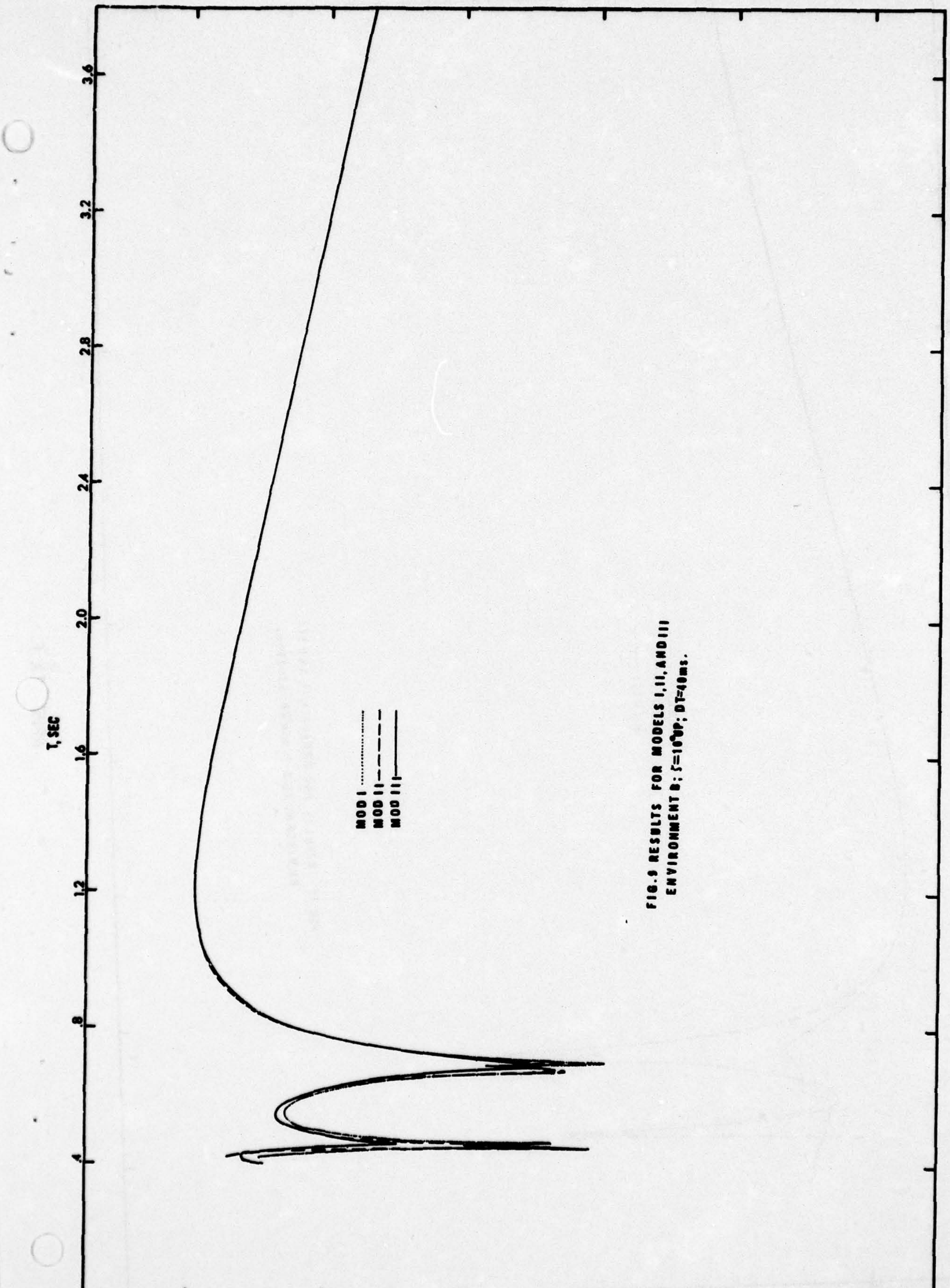


FIG.9 RESULTS FOR MODELS I,II,AND III
ENVIRONMENT B; ζ=10°UP; DT=40ms.

APPENDIX F

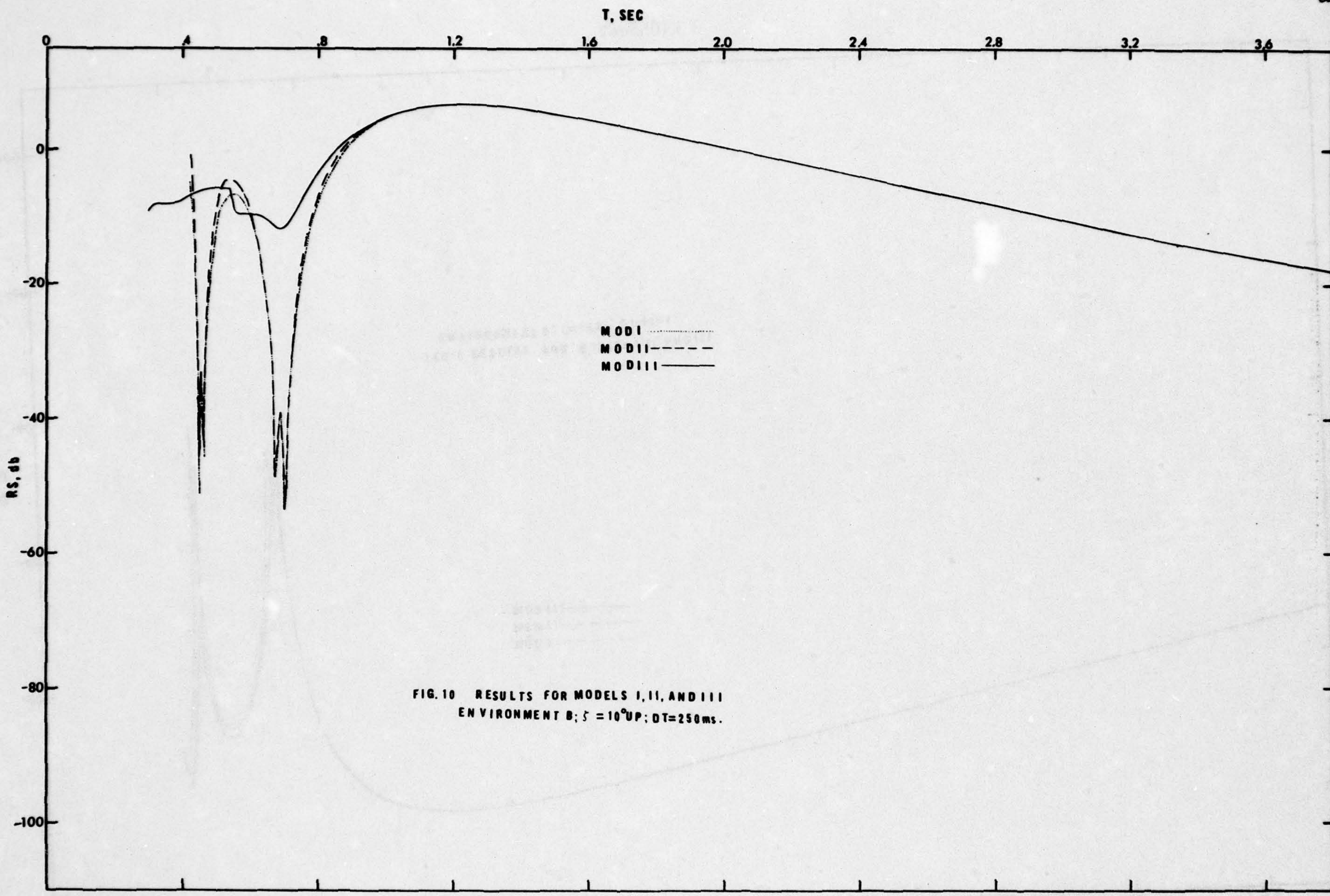


FIG. 10 RESULTS FOR MODELS I, II, AND III
ENVIRONMENT B; ζ = 10° UP; DT=250ms.

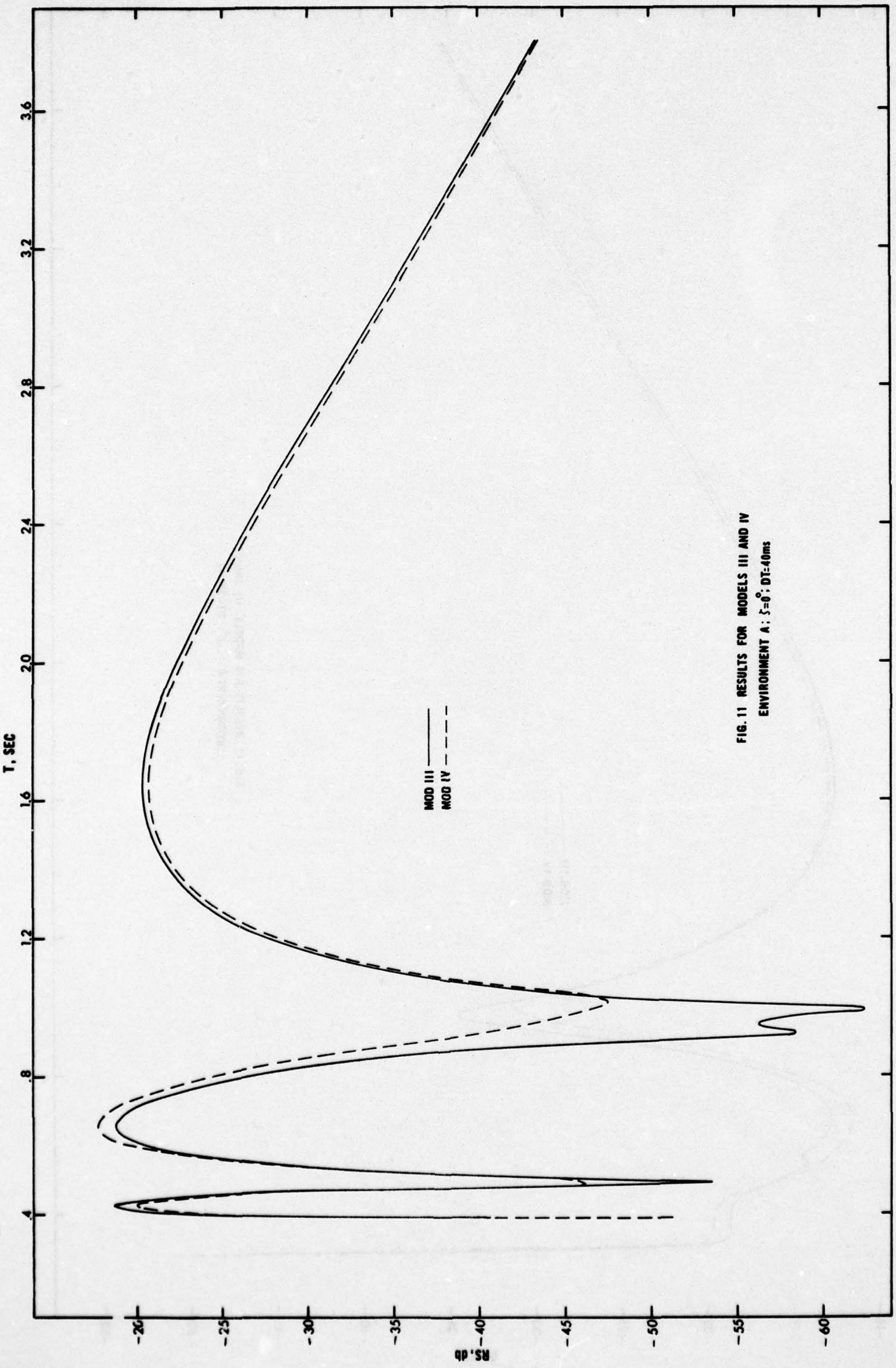


FIG. 11 RESULTS FOR MODELS III AND IV
ENVIRONMENT A; $\zeta=0^{\circ}$; DT=40ms

APPENDIX F

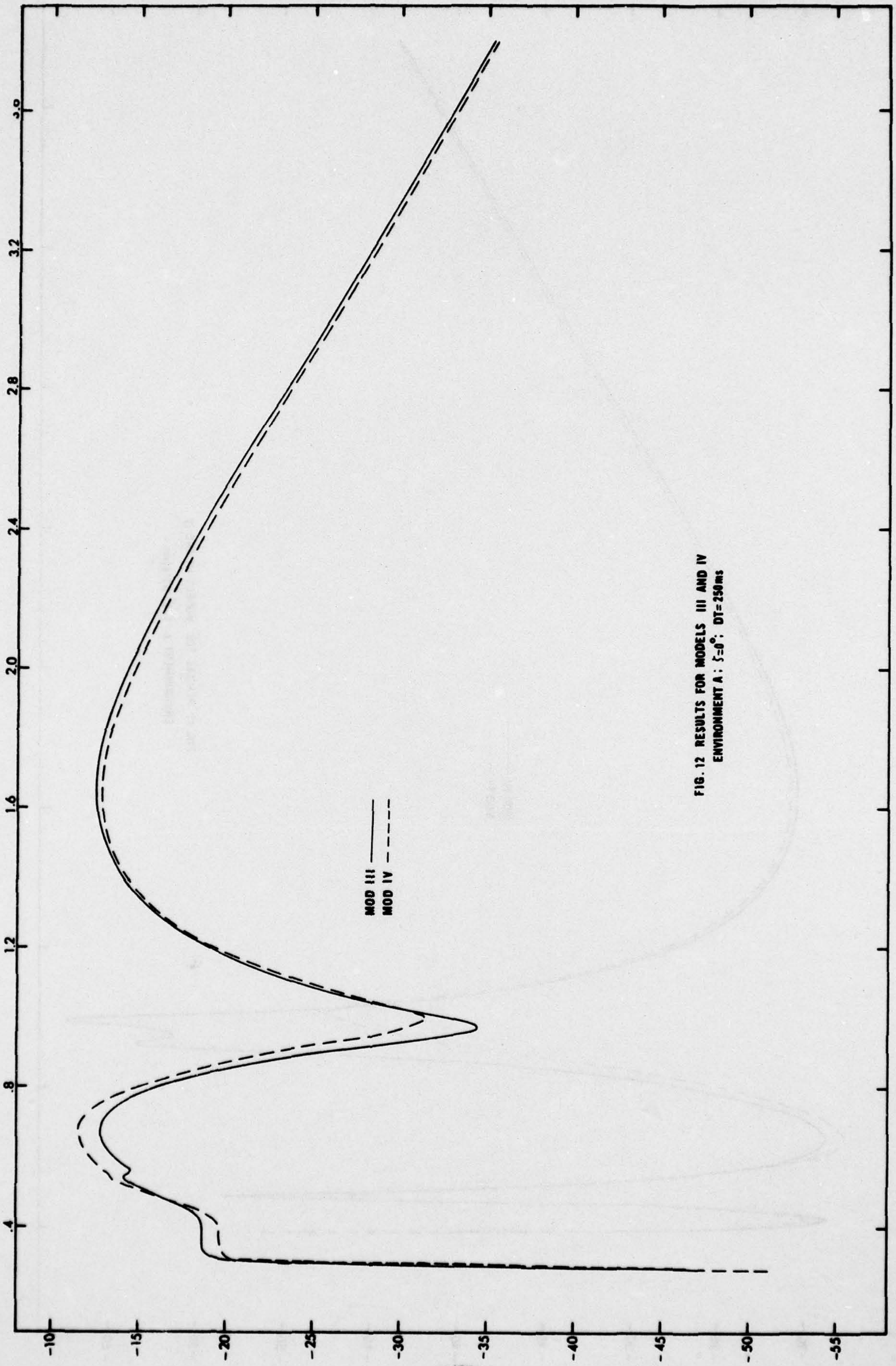


FIG. 12 RESULTS FOR MODELS III AND IV
ENVIRONMENT A; $\zeta=0^{\circ}$; DT=250ms

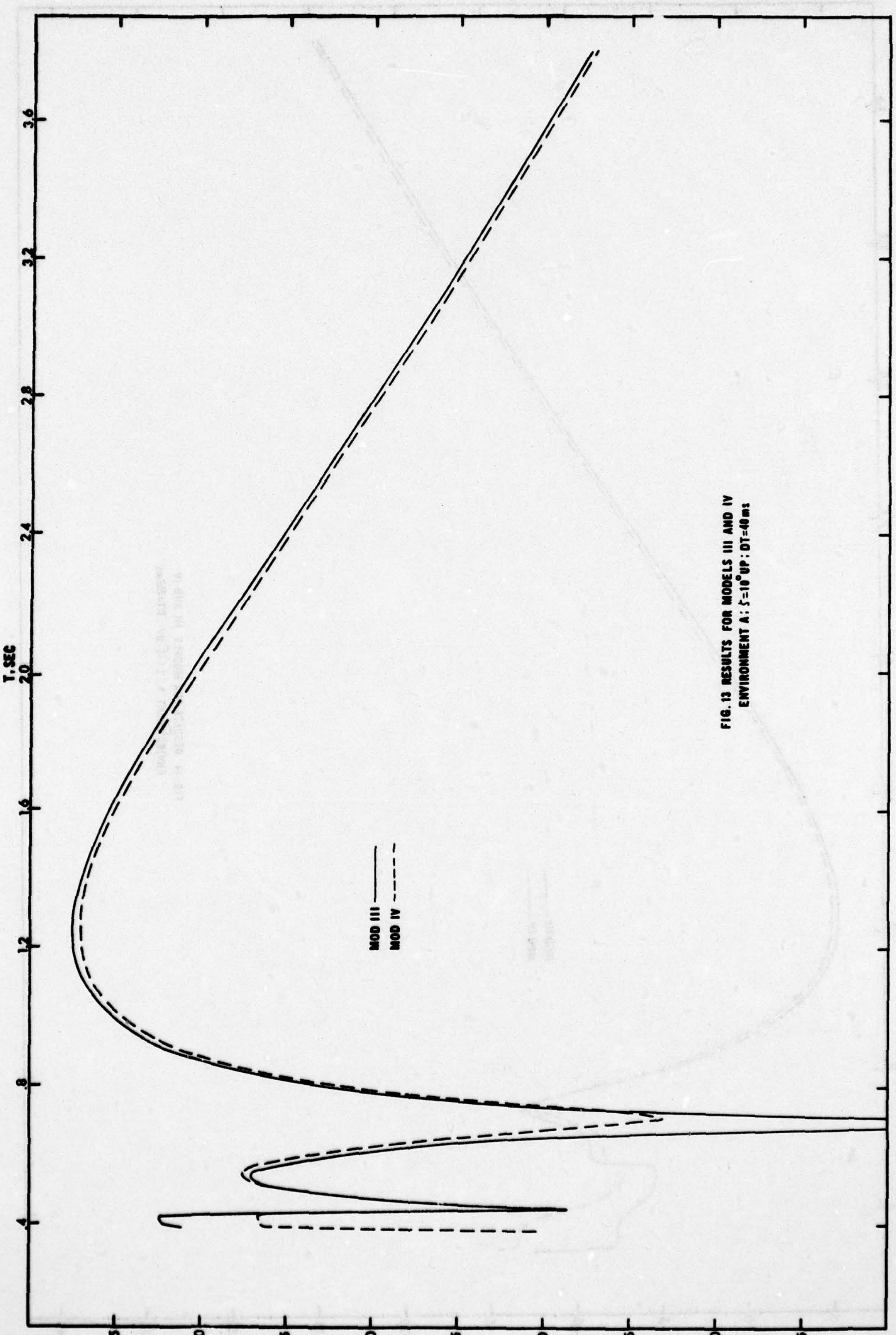


FIG. 13 RESULTS FOR MODELS III AND IV
ENVIRONMENT A: ζ=10° UP; DT=40ms

APPENDIX F

OD 52258

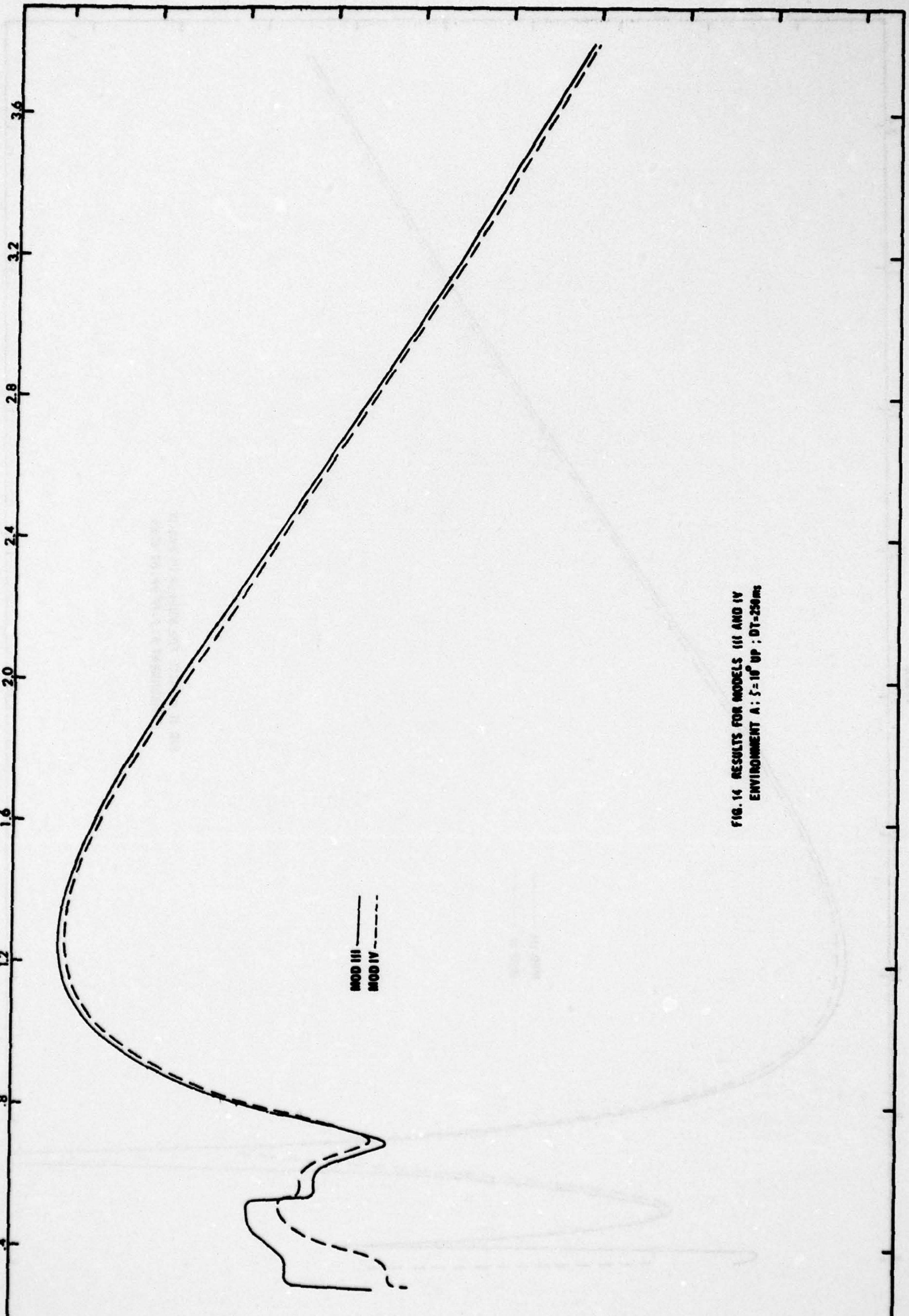


FIG. 14 RESULTS FOR MODELS III AND IV
ENVIRONMENT A; $\zeta = 10°$ UP; DT=250ms

APPENDIX F

AD-A035 071 NAVAL SEA SYSTEMS COMMAND WASHINGTON D C F/G 17/1

A COMPUTER PROGRAM FOR STUDYING THE DOPPLER CONTENT OF REVERBER--ETC(U)

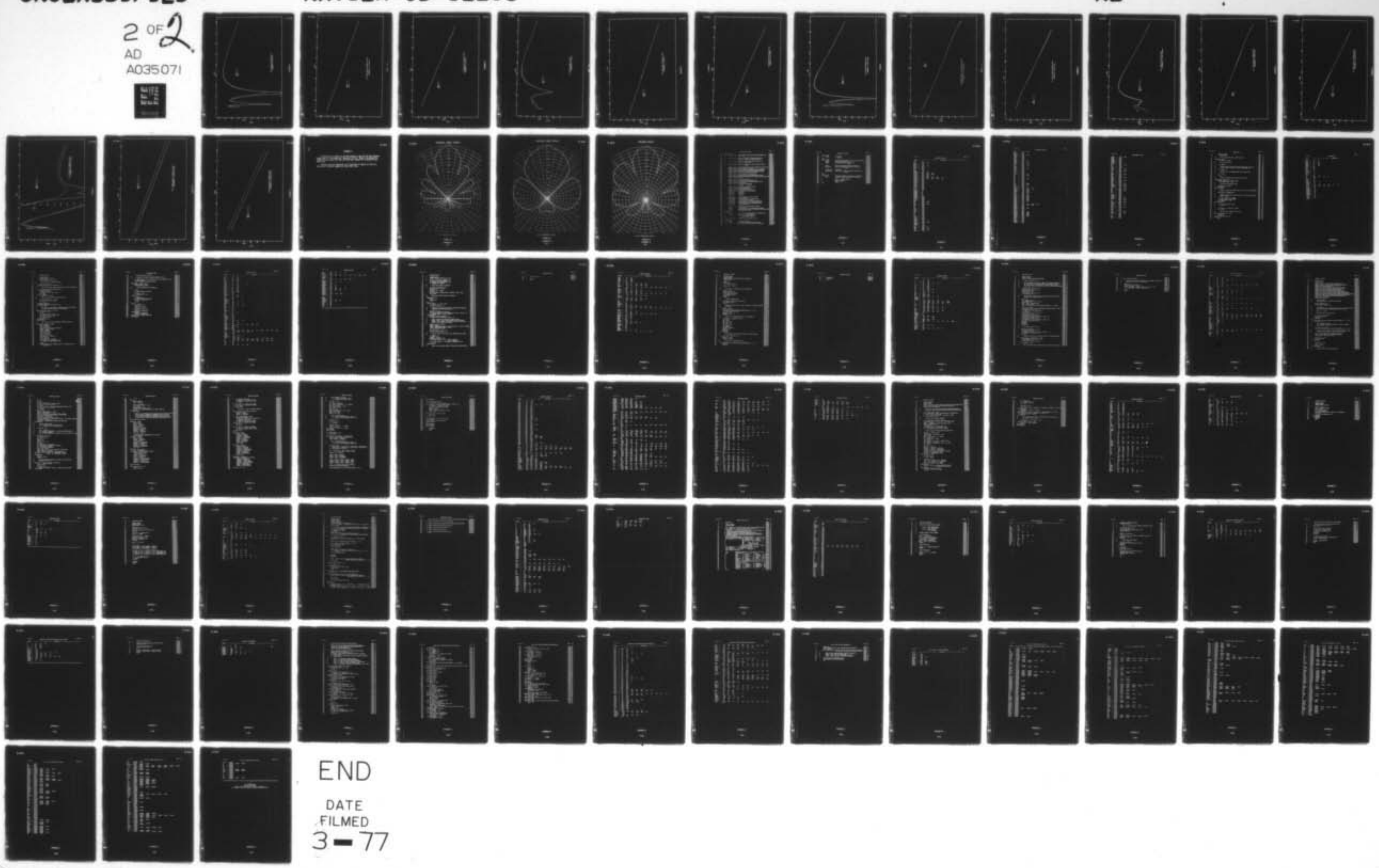
1976 P MARSH

UNCLASSIFIED NAVSEA-OD-52258 NL

2 OF 2

AD

A035071



END

DATE

FILMED

3-77

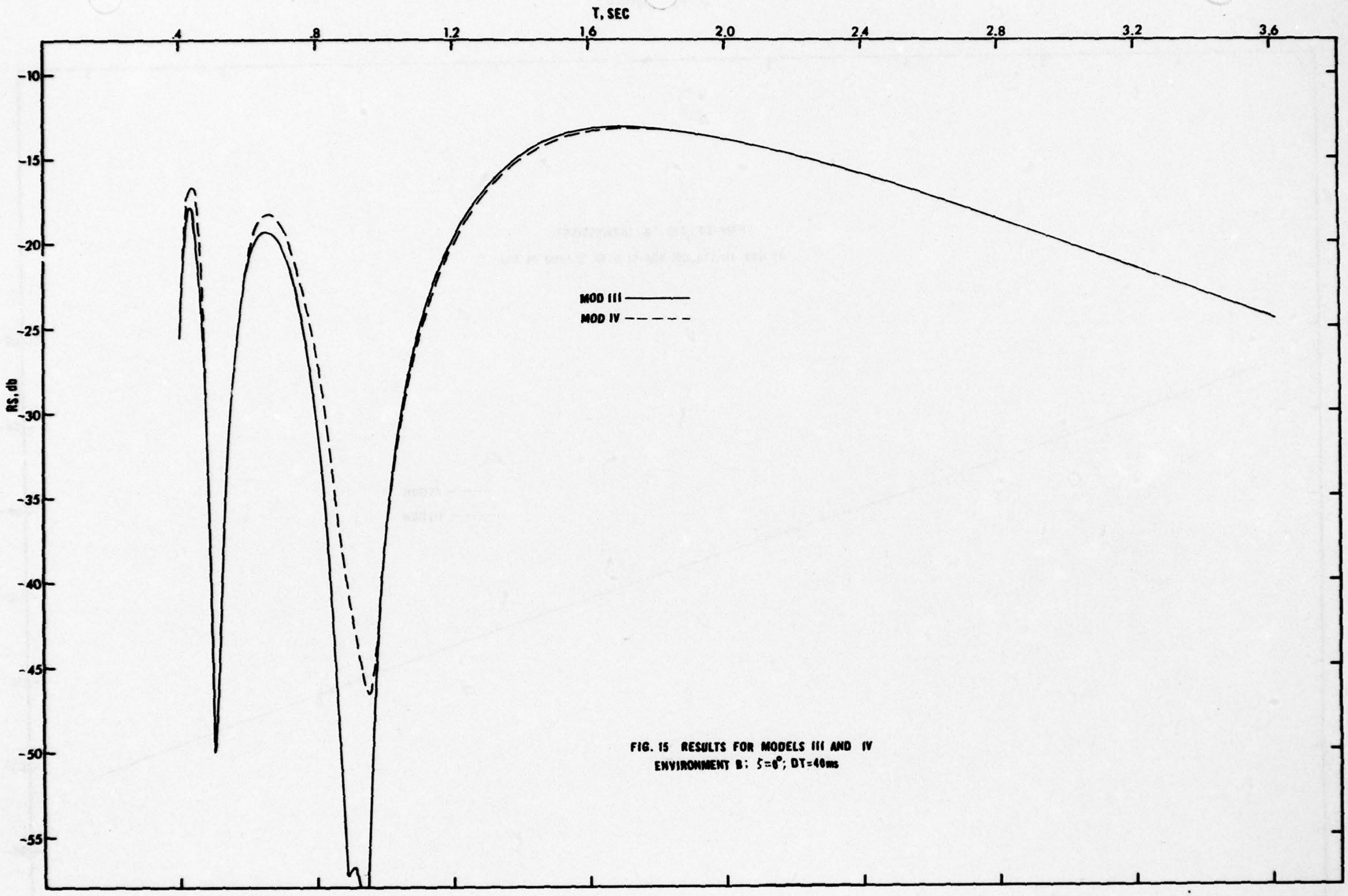


FIG. 15 RESULTS FOR MODELS III AND IV
ENVIRONMENT B; $\zeta=0°$; DT=40ms

APPENDIX F

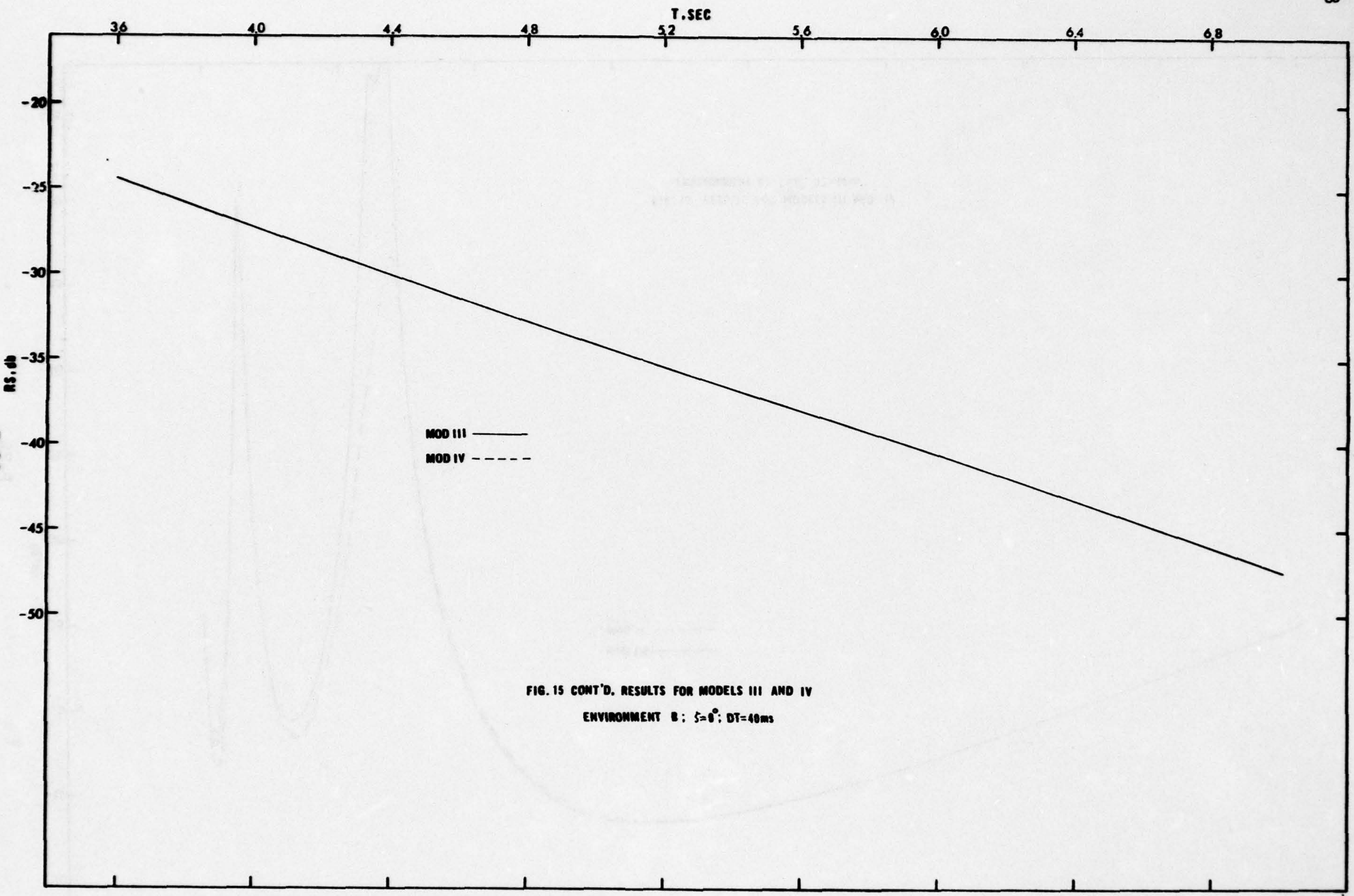


FIG. 15 CONT'D. RESULTS FOR MODELS III AND IV

ENVIRONMENT B ; $\zeta=0^{\circ}$; DT=40ms

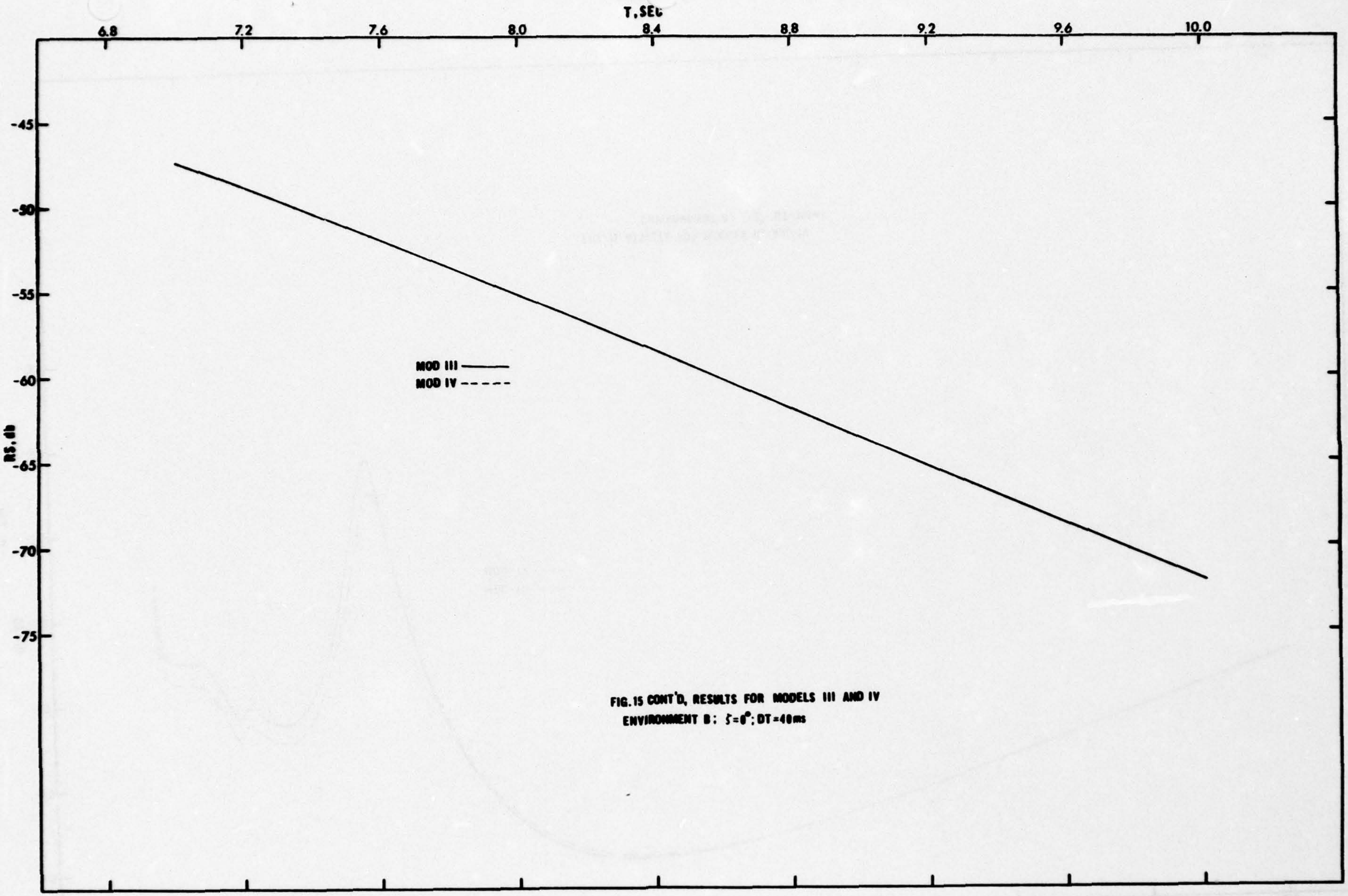


FIG. 15 CONT'D, RESULTS FOR MODELS III AND IV
ENVIRONMENT B: $\zeta = 0^{\circ}$; DT = 40ms

APPENDIX F

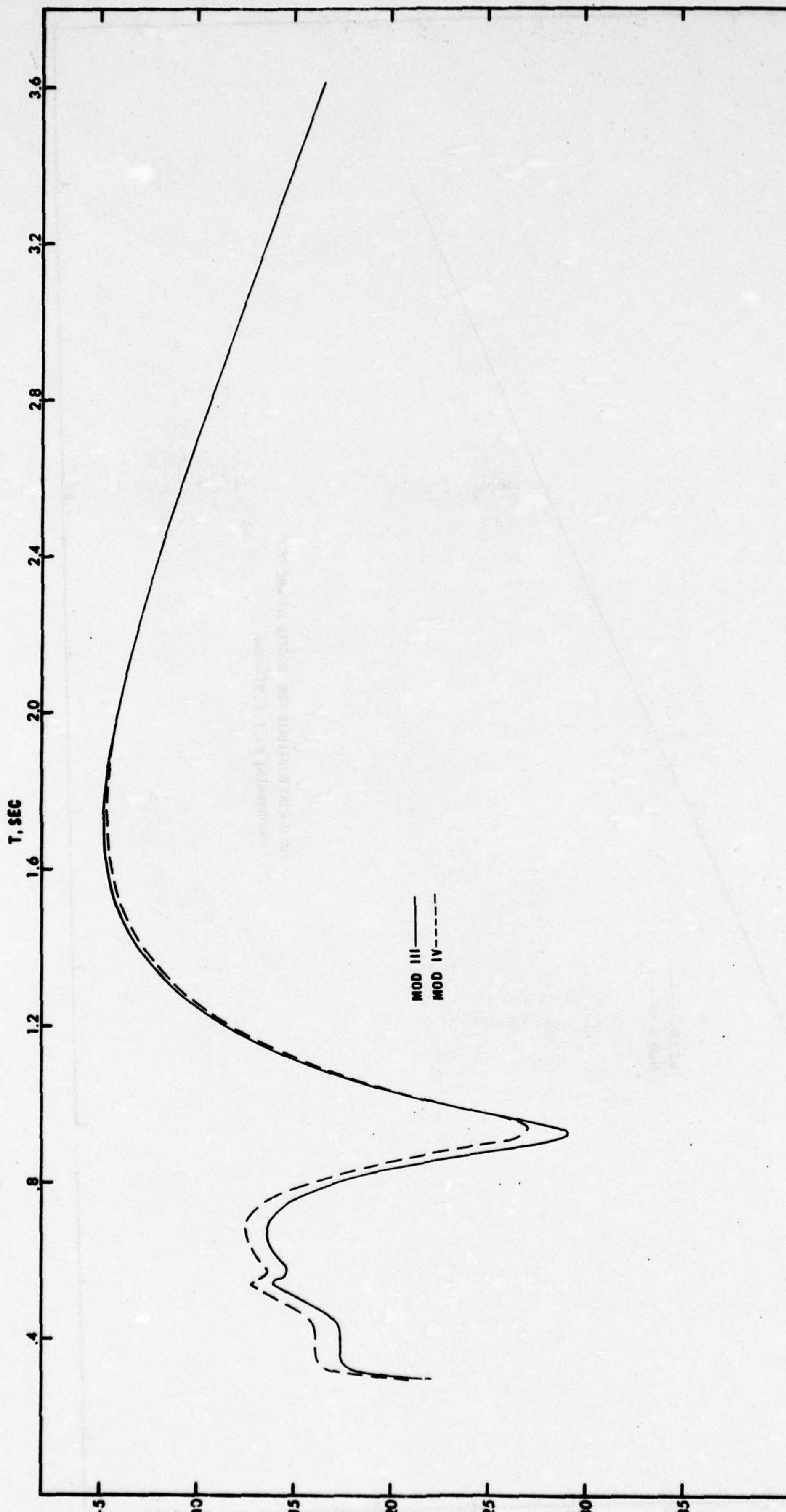


FIG. 18 RESULTS FOR MODELS III AND IV
ENVIRONMENT B; $\zeta = 0^{\circ}$ DT=250ms

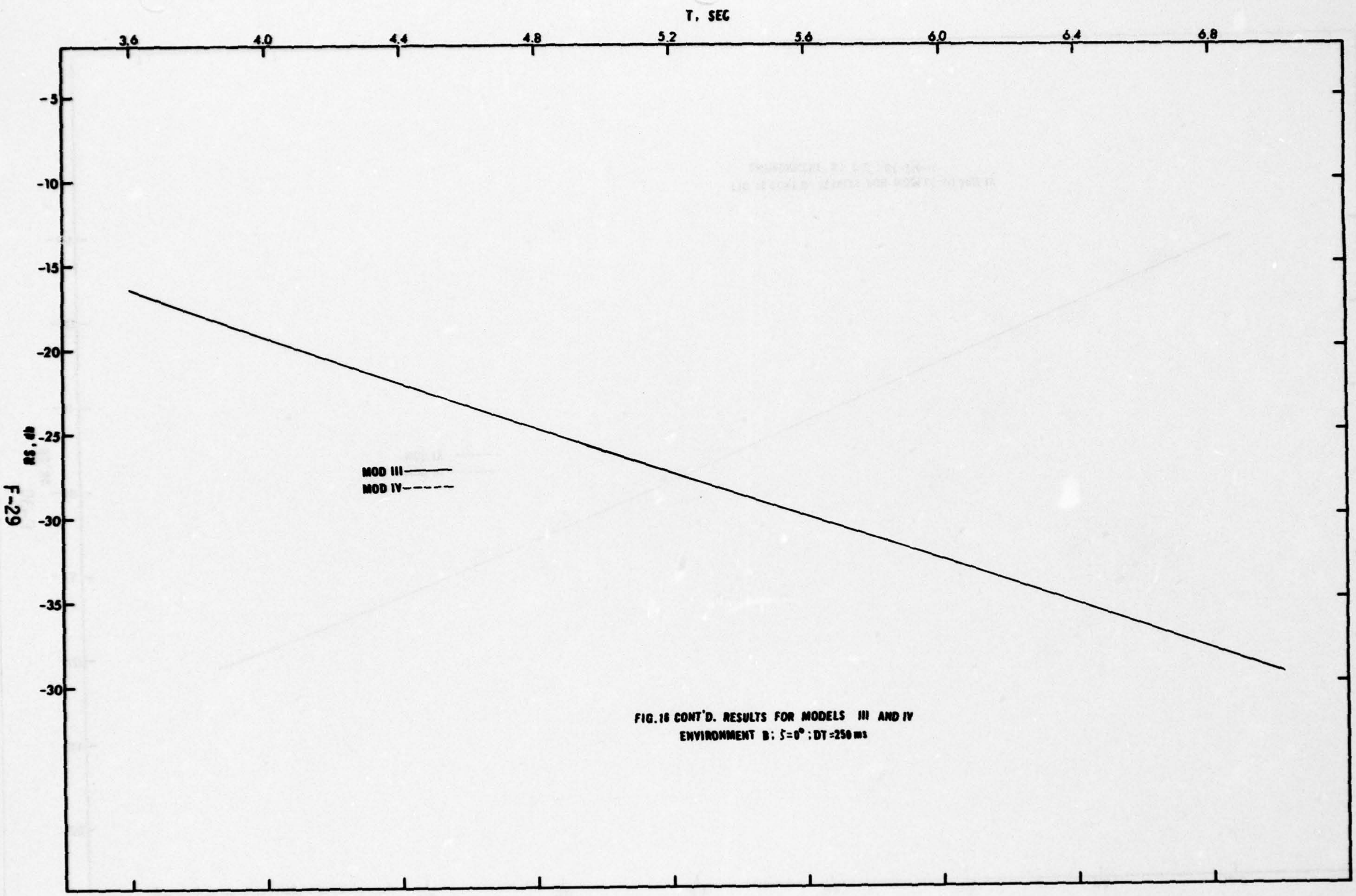


FIG.18 CONT'D. RESULTS FOR MODELS III AND IV
ENVIRONMENT B; $\zeta=0°$; DT=250 ms

APPENDIX F

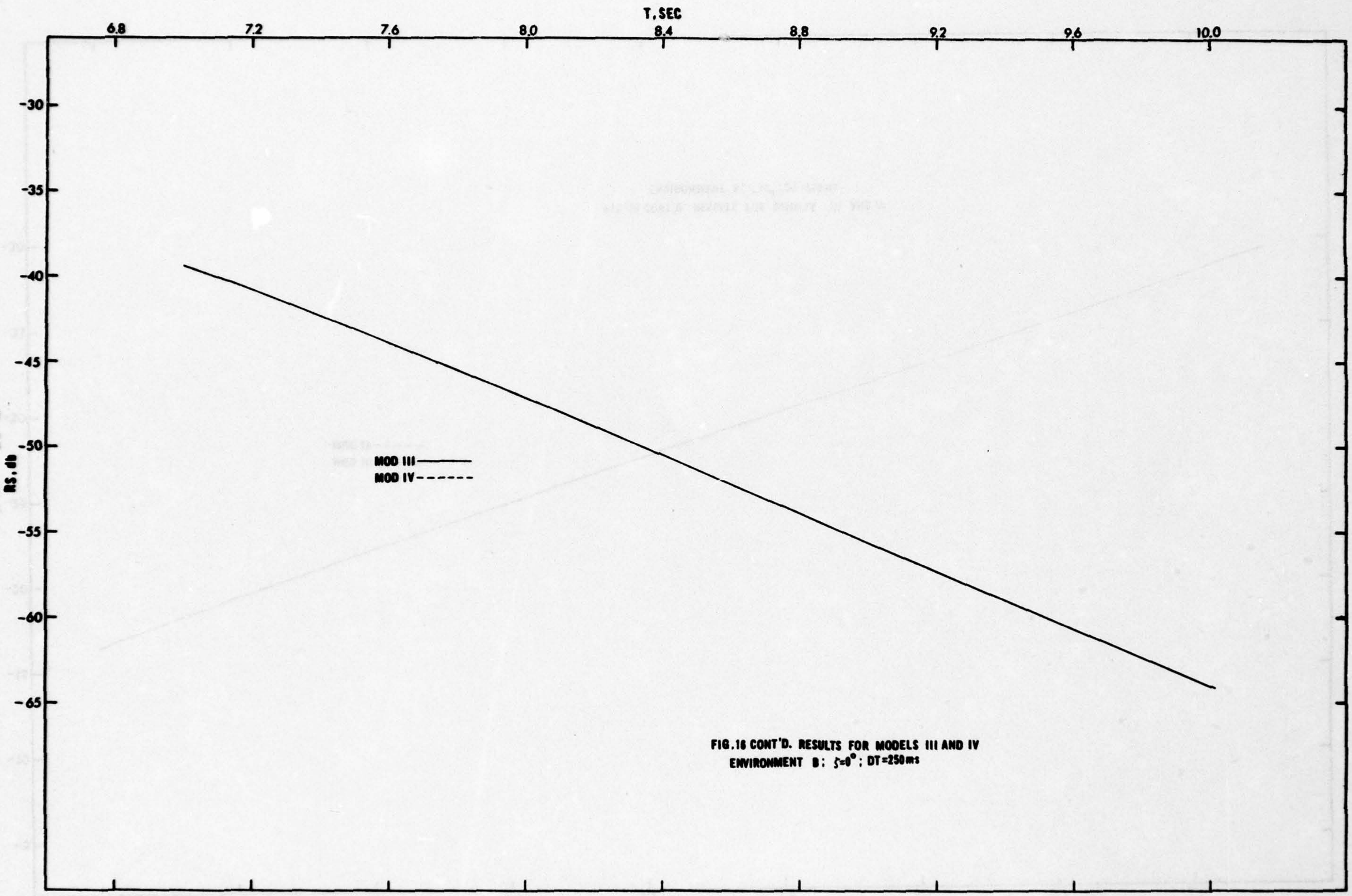


FIG. 16 CONT'D. RESULTS FOR MODELS III AND IV
ENVIRONMENT B: $\zeta=0^{\circ}$; DT=250ms

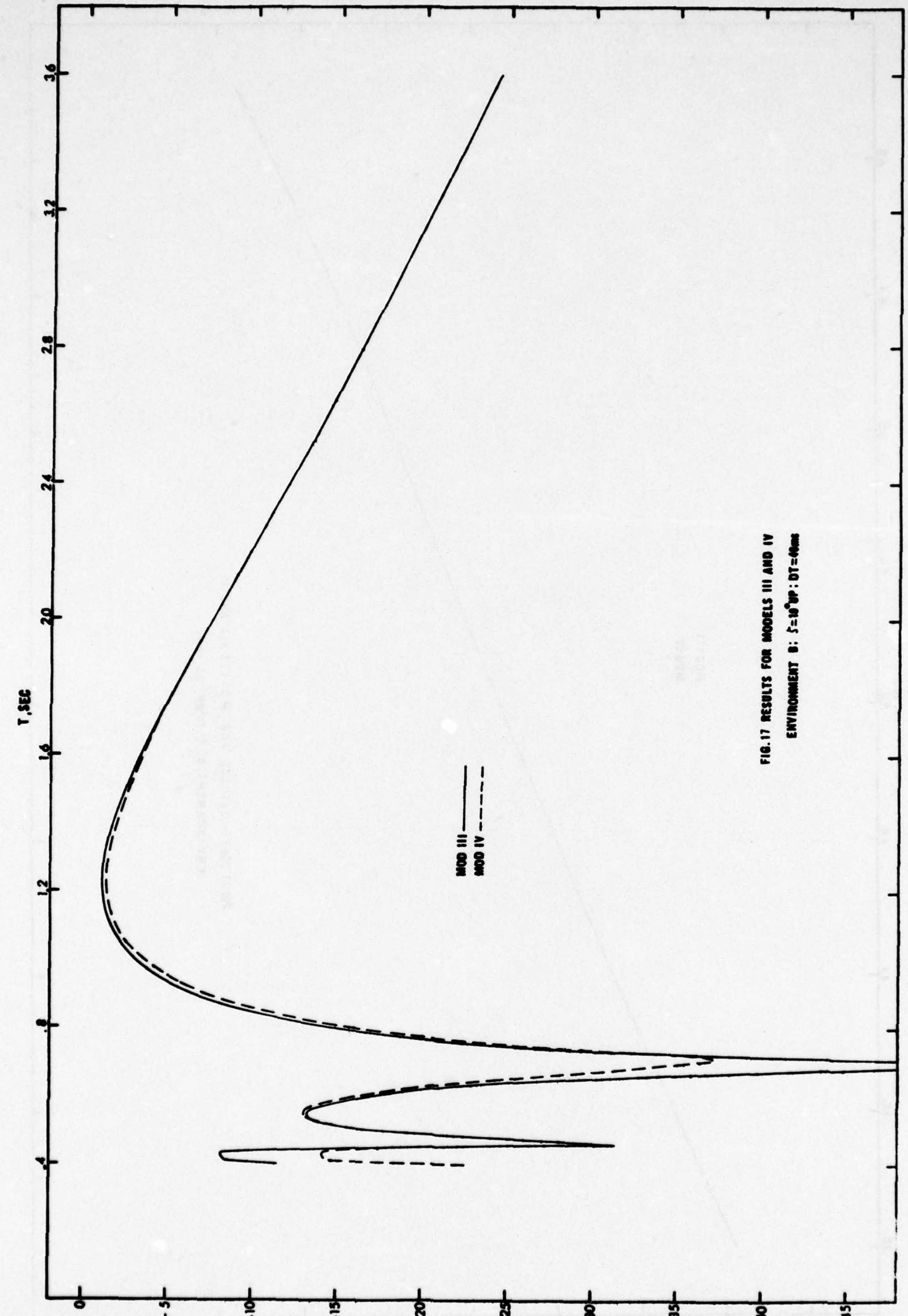


FIG.17 RESULTS FOR MODELS III AND IV
ENVIRONMENT B: $\zeta=10^\circ$UP: DT=40ms

APPENDIX

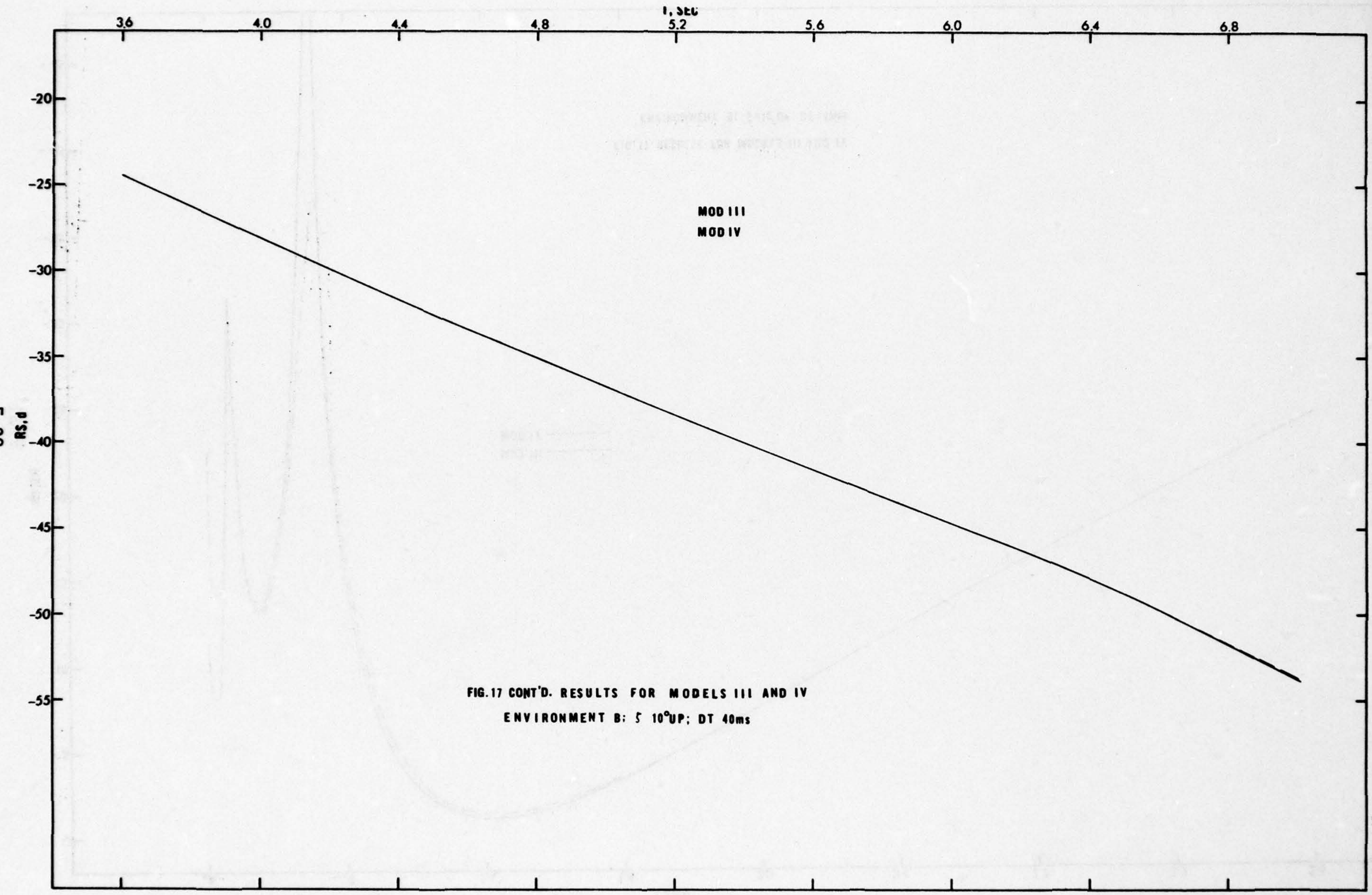


FIG.17 CONT'D. RESULTS FOR MODELS III AND IV
ENVIRONMENT B: S 10°UP; DT 40ms

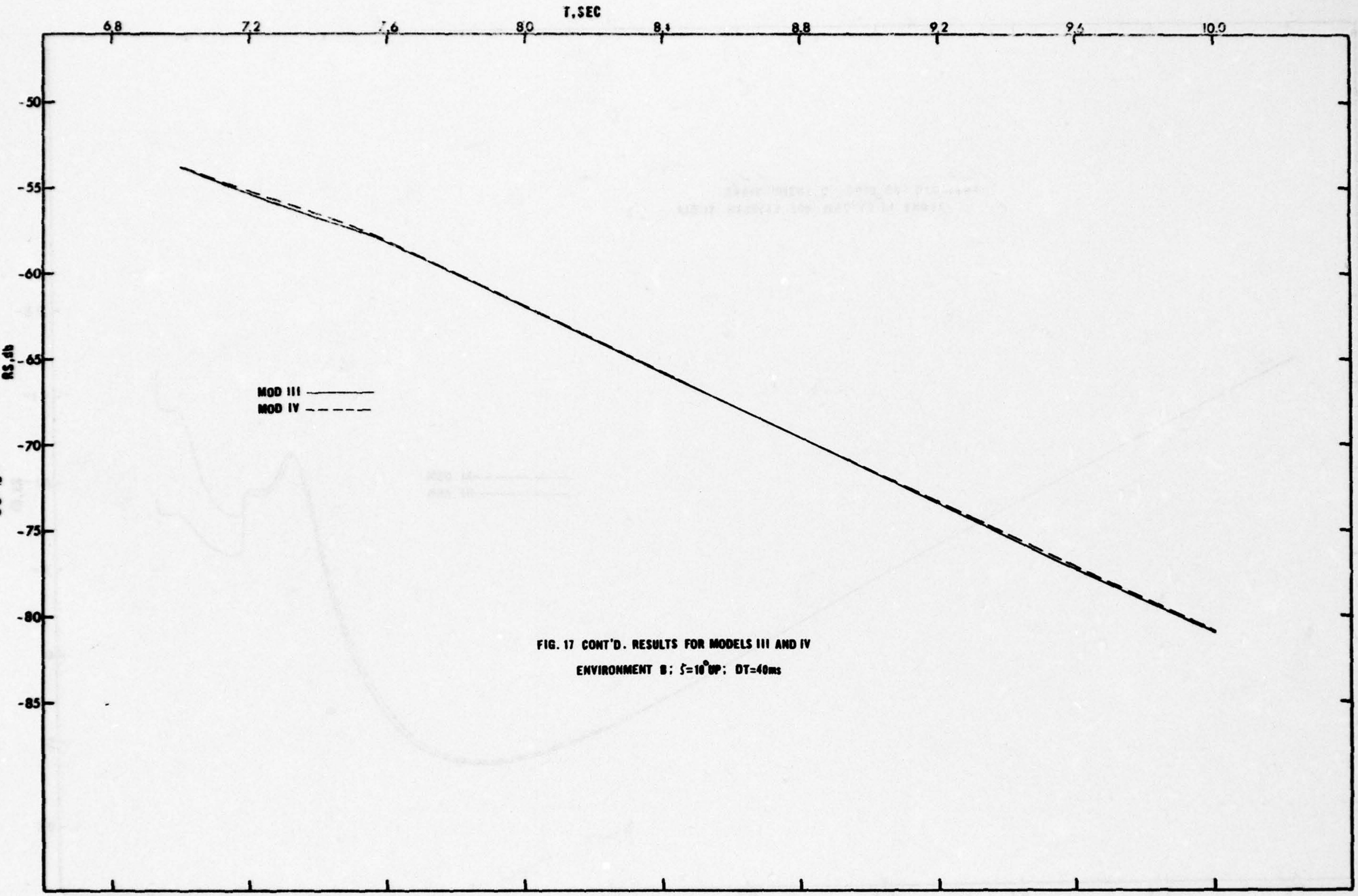


FIG. 17 CONT'D. RESULTS FOR MODELS III AND IV

ENVIRONMENT B; $\zeta=10^{\circ}$UP; DT=40ms

OD 52258

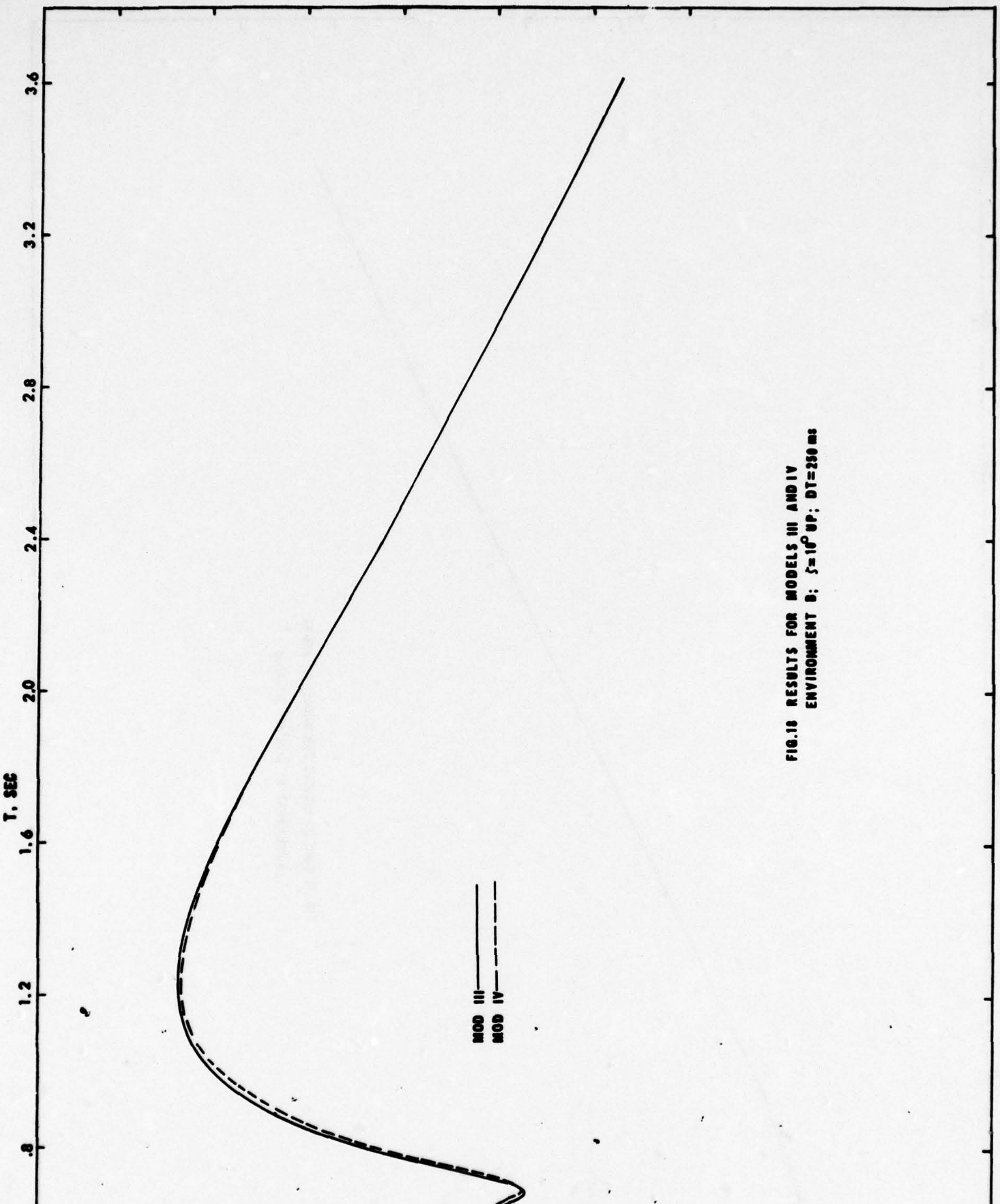


FIG.18 RESULTS FOR MODELS III AND IV
ENVIRONMENT B; $\zeta=10^{o}$ UP; DT=250 ms

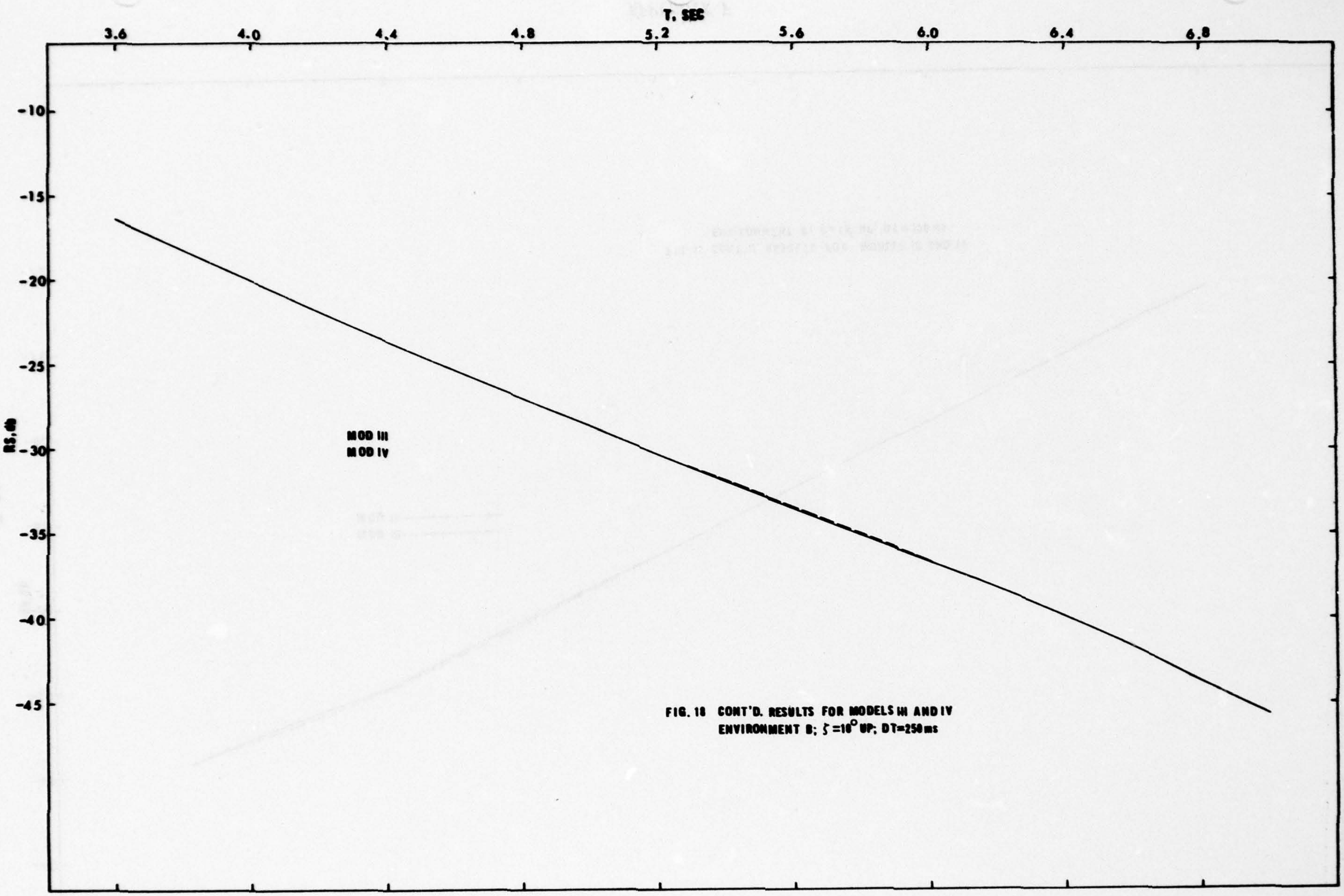


FIG. 18 CONT'D. RESULTS FOR MODELS III AND IV
ENVIRONMENT B; $\zeta = 10^{\circ}$ UP; DT=250ms

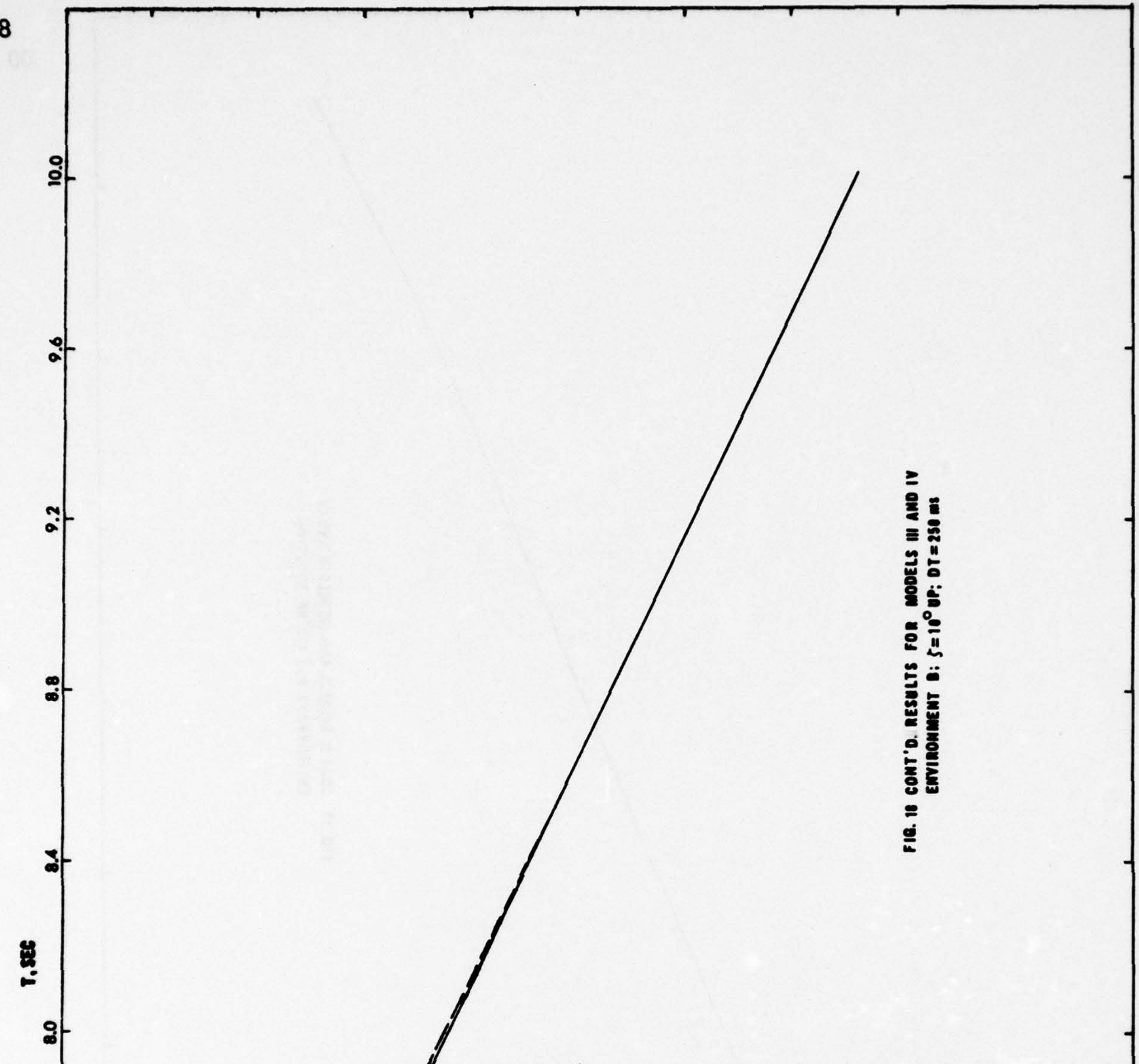


FIG. 18 CONT'D. RESULTS FOR MODELS III AND IV
ENVIRONMENT B; $\zeta = 10^\circ$ UP; DT=250 ms

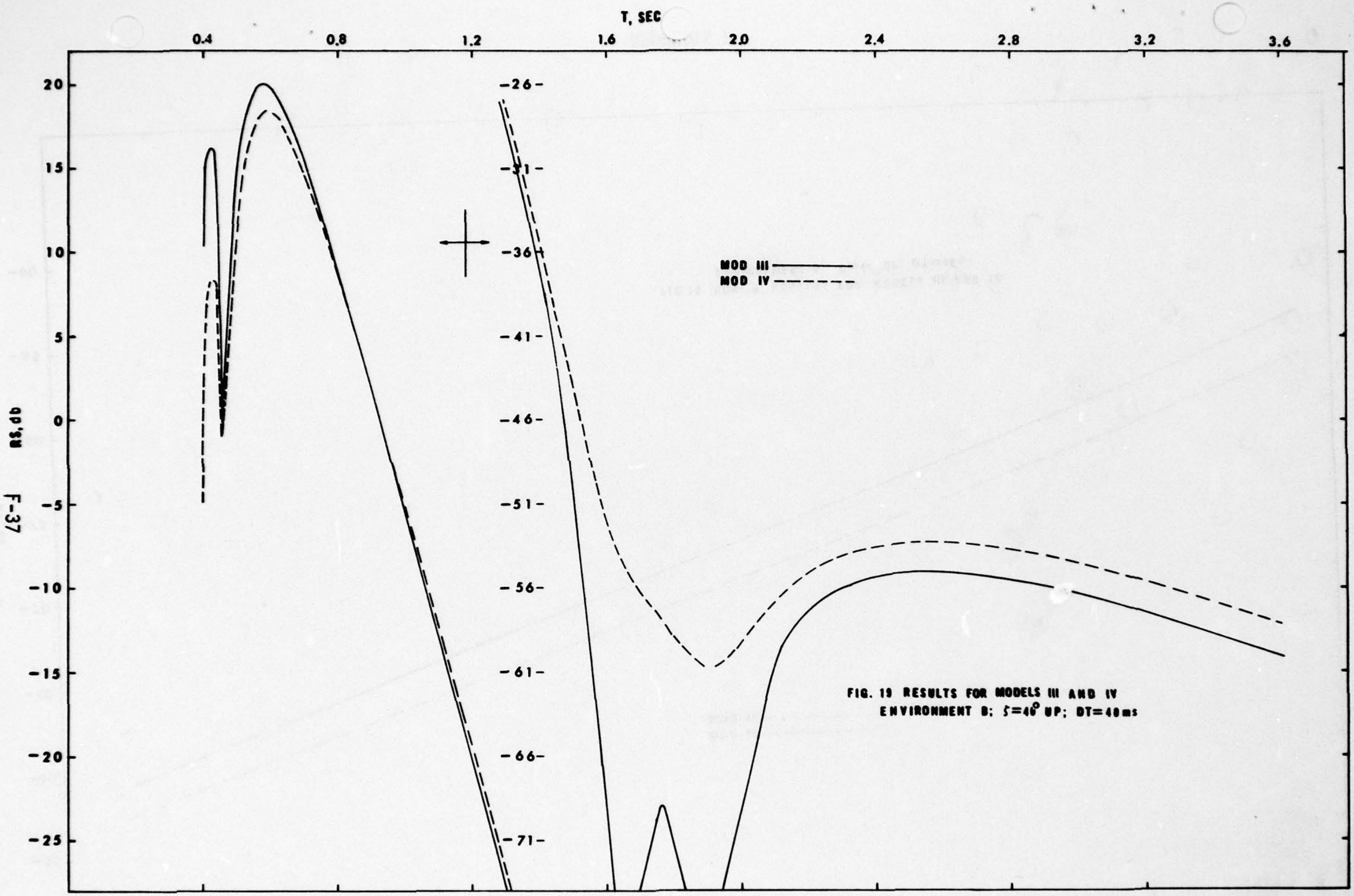


FIG. 19 RESULTS FOR MODELS III AND IV
ENVIRONMENT B; $\zeta=40^{\circ}$ UP; DT=40ms

OD 52258

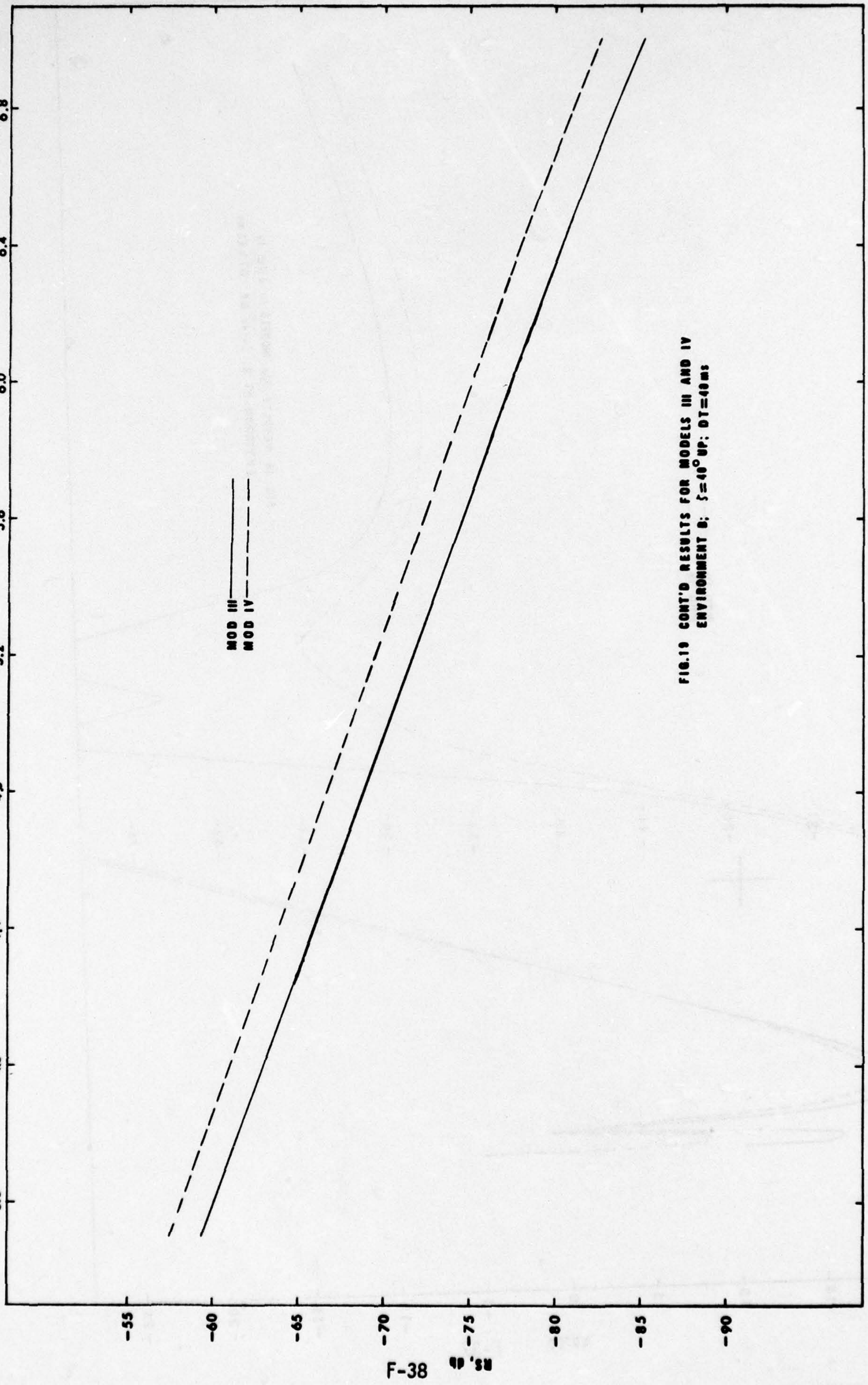


FIG.19 CONT'D RESULTS FOR MODELS III AND IV
ENVIRONMENT B; $\zeta=40^{\circ}$ UP; DT=40ms

APPENDIX F

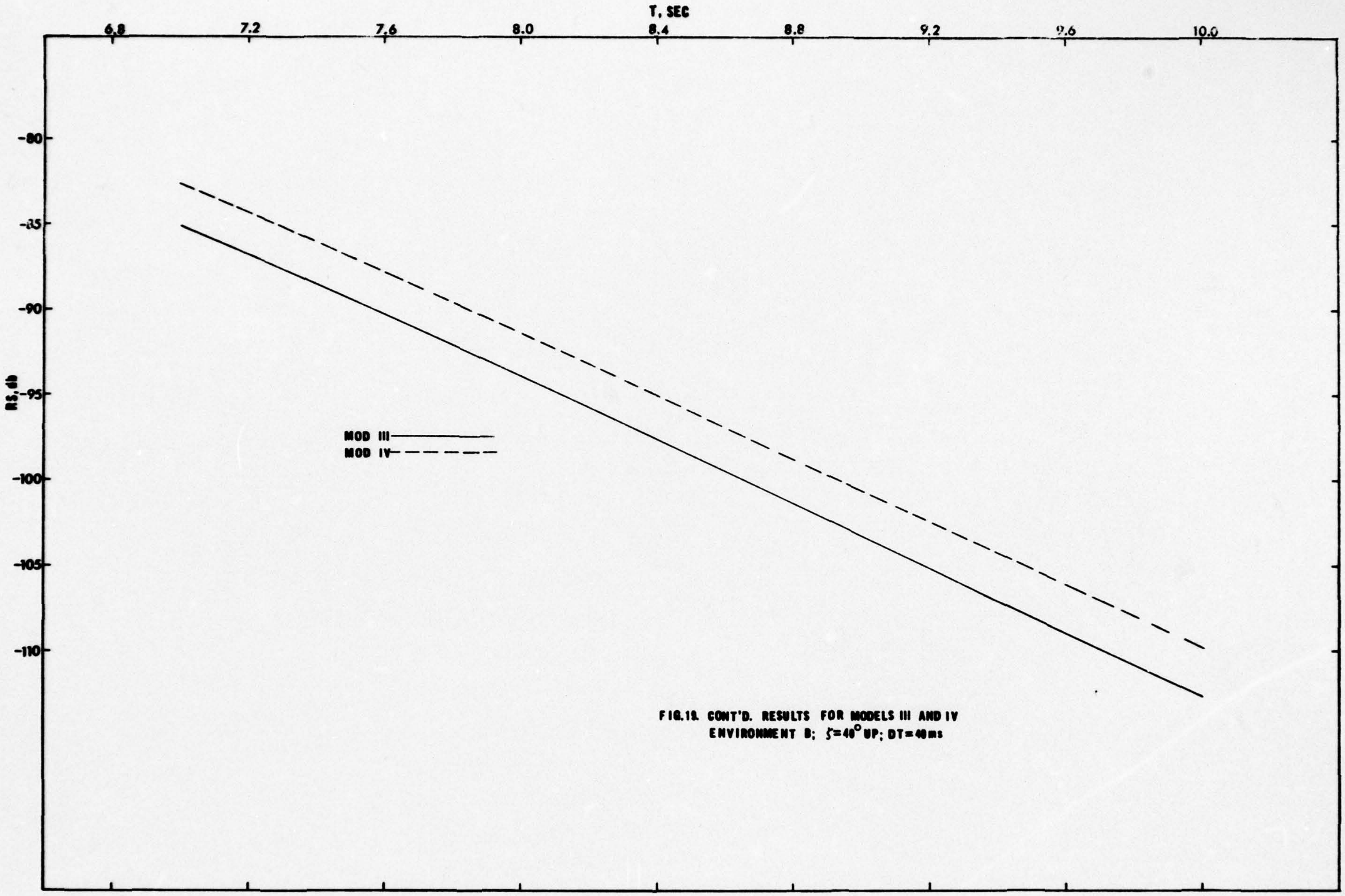


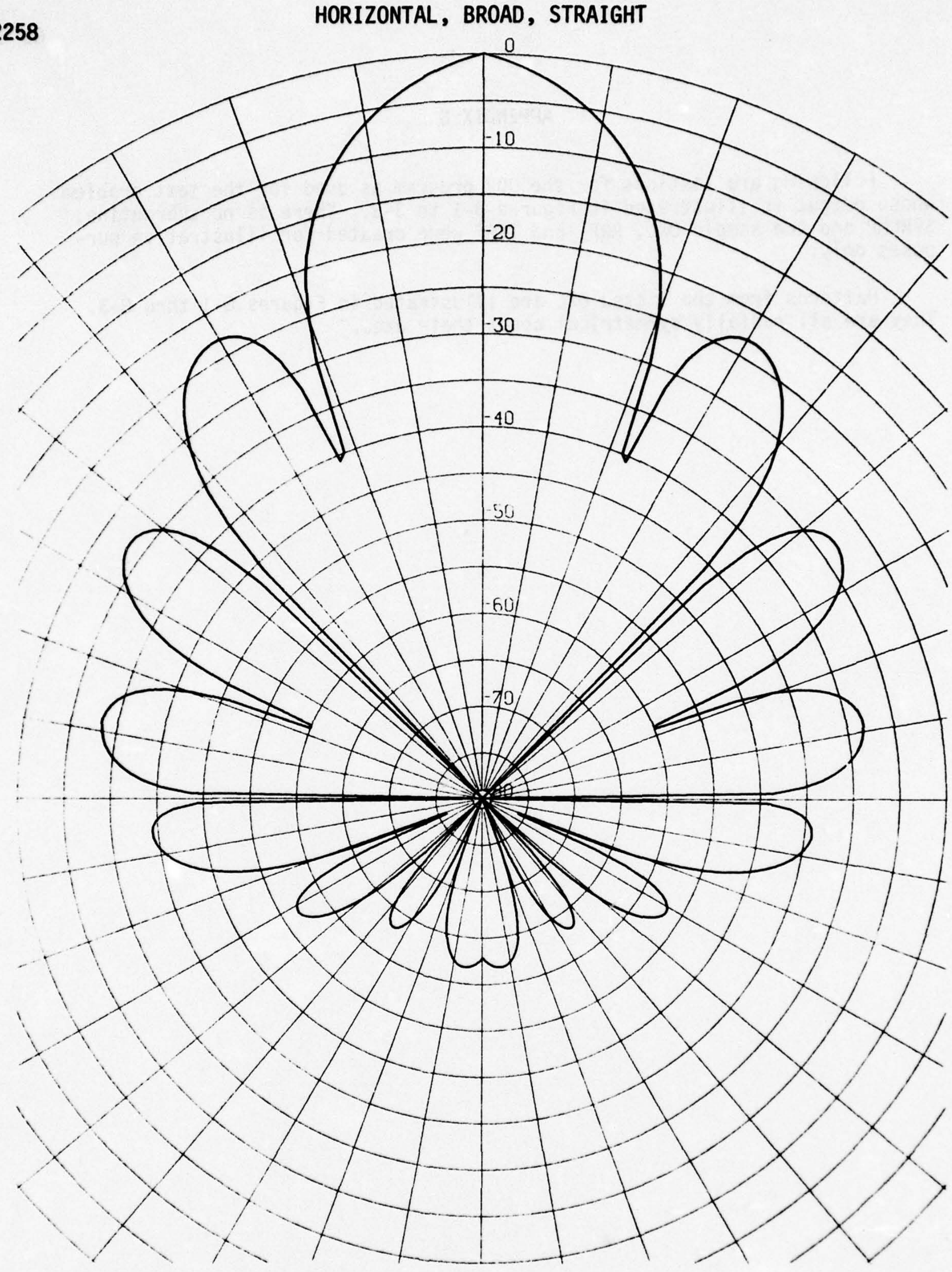
FIG.19. CONT'D. RESULTS FOR MODELS III AND IV
ENVIRONMENT B; ζ=40° UP; DT=40ms

APPENDIX G

Following are listings for the DOP program as used for the test problem whose output is illustrated in Figures 3-1 to 3-3. There is no subroutine, SPRCMP and the sample OXL, RRF, and TVGF were created for illustrative purposes only.

Patterns from the listed OXL are illustrated in Figures G-1 thru G-3. They are all radially symmetrical about their axes.

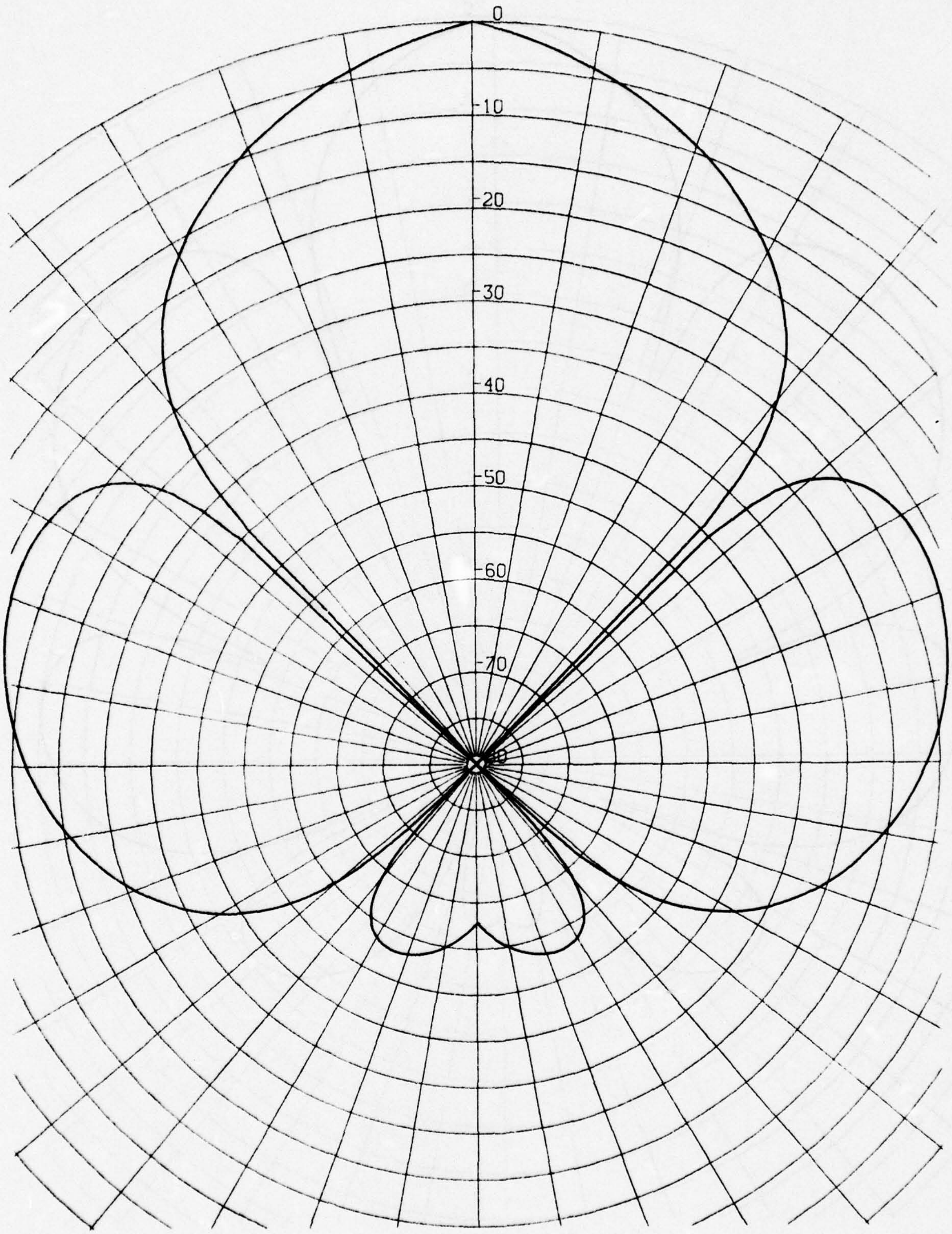
OD 52258



AXIAL COORDINATE PLANES

FIGURE G-2

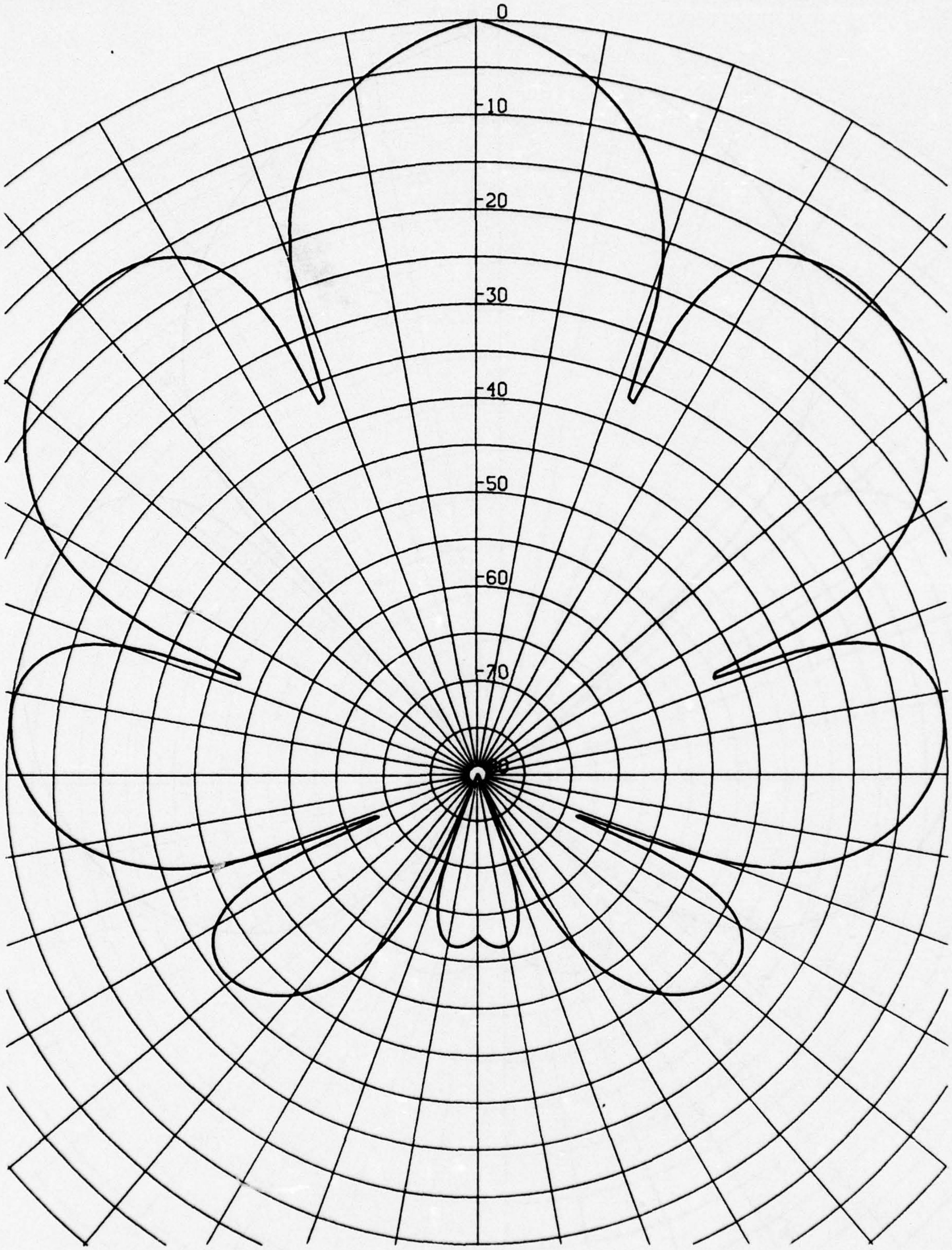
HORIZONTAL, BROAD, STRAIGHT



AXIAL COORDINATE PLANES

FIGURE G-2

HORIZONTAL RECEIVE



AXIAL COORDINATE PLANES

FIGURE G-1

PROCEDURES FOR DOP

```
DCOMN1  PROC                                                            DCOMMN 1
      COMMON /CBAND/  LMBAND,LMBND1,NBAND,MBAND,MNBND,FNBAND,VPTTRN     DCOMMN 2
C                      800  , 801                                       DCOMMN 3
C                                                                       DCOMMN 4
      COMMON /CBAND/  BAND(801),RRFS(800),FGAM(800),BNDOUT(801)         DCOMMN 5
C                     (LMBND1) ,(LMBAND) ,(LMBAND),(LMBND1)             DCOMMN 6
C                                                                       DCOMMN 7
      COMMON /CCOUNT/ LMRB,LMTRS,LMKS,LMKS2,NSURF,NBTM,KT,KTT           DCOMMN 8
C                     9612, 17  , 400, 800                              DCOMMN 9
C                                                                       DCOMMN10
      COMMON /CFCNST/ TRS(6),FPT5,TRS2(4),F1,TRS3(4),F2,F4,F10,F20      DCOMMN11
C                   (           LMTRS                  )                DCOMMN12
C                                                                       DCOMMN13
      COMMON /CFCNST/ FLO610,INFNT,P123,PI,TWOPI,DEGRAD,SHIFT,VDKT,F1E3 DCOMMN14
      COMMON /CFCNST/ LO64PI,F1MIN,F3,F90,F180                          DCOMMN15
      COMMON /CHCNST/ HBAND(3),HUNIT(2),HSPRD(5),HSPRV(5),HFROM(2)      DCOMMN16
      COMMON /CHCNST/ HEADS(41),HTOT1(4),HTOT2(3),HTOT3(3),HED(9)       DCOMMN17
      COMMON /CINDAT/ OMEGA,DELT,CSKSI,SNKSI,KSID,DR,FCOVS,COKT         DCOMMN18
      COMMON /CINDAT/ EXPS,F2SQ,F22,FC03,BWINT,LOGMVI                   DCOMMN19
C                                                                       DCOMMN20
      COMMON /CINDEF/ NNAMES,NAMDAT(40,3),NAMCNT(24)                    DCOMMN21
      COMMON /CINDEF/ CENTER,GO,END,FILTER,KTSBND,NOBTTM,NOPRNT         DCOMMN22
      COMMON /CINDEF/ NOSURF,NOTAPE,NOVOLM,PLOT,SPREAD,TIMCMP,TOTALS,TVGDCOMMN23
      COMMON /CINDEF/ RELBND                                            DCOMMN24
C                                                                       DCOMMN25
      COMMON /CINPUT/ IDC,IDATE(2),IDV,DO                               DCOMMN26
      COMMON /CINPUT/ CO,ALPHC,PING,DRTTM,LOGMV,S,KSI                   DCOMMN27
      COMMON /CINPUT/ VS,BWIDTH,DELT2,FZRO,NBEAM,OMEGAD,THTMAX          DCOMMN28
      COMMON /CINPUT/ PULSE,IPEVRY                                      DCOMMN29
      COMMON /CINPUT/ TIME(400),SPRED(150,3)                            DCOMMN30
C                       (LMTIM)      (LMSPRD)                           DCOMMN31
C                                                                       DCOMMN32
      COMMON /CPRINT/ NTMIN,NTMAX,PAGE,NPAGE,NPSTRT                     DCOMMN33
      COMMON /CSPRED/ NSPR1,NSPRH,NSPRH1                                DCOMMN34
      COMMON /CTAPE/  AR1,BR1,INPT,IPRT,IPLT                            DCOMMN35
      COMMON /CTBLKP/ DELIND,DELDEP,FACTOR,ITABLE                       DCOMMN36
      COMMON /CXCNST/ K0,K1,K2,K3,K5,K6,K8,K10,K40                      DCOMMN37
      DIMENSION       NBOUND(2)                                         DCOMMN38
      DIMENSION       HOUTPT(14)                                        DCOMMN39
      DIMENSION       IDATA(1)                                          DCOMMN40
      DIMENSION       LFLAGS(16)                                        DCOMMN41
C                                                                       DCOMMN42
      EQUIVALENCE     (K0,F0,RECV),(K1,XMIT),(K3,LMNT)                  DCOMMN43
      EQUIVALENCE     (NSURF,NBOUND)                                    DCOMMN44
      EQUIVALENCE     (IBLANK,HEADS(16)),(HUNIT,HOUTPT)                 DCOMMN45
      EQUIVALENCE     (IFZRO,FZRO),(IDC,IDATA)                          DCOMMN46
      EQUIVALENCE     (LMTIM,NAMDAT(21,3)),(NTIME,NAMCNT(21))           DCOMMN47
      EQUIVALENCE     (LMSPRD,NAMDAT(22,3)),(LFLAGS,CENTER)             DCOMMN48
      EQUIVALENCE     (NSSPRD,NAMCNT(22)),(NBSPRD,NAMCNT(23))           DCOMMN49
      EQUIVALENCE     (NVSPRD,NAMCNT(24))                               DCOMMN50
C                                                                       DCOMMN51
      REAL            INFNT,KSI,KSID,LOGMV,LO64PI,LOGMVI                DCOMMN52
      INTEGER         AR1,BR1,PAGE                                      DCOMMN53
      LOGICAL         CENTER,GO,END,FILTER,KTSBND,NOBTTM,NOPRNT         DCOMMN54
      LOGICAL         NOSURF,NOTAPE,NOVOLM,PLOT,SPREAD,TIMCMP,TOTALS,TVGDCOMMN55
      LOGICAL         RELBND,VPTTRN,LFLAGS                              DCOMMN56
 END                                                                    DCOMMN57
DCOMN2  PROC                                                            DCOMMN58
      COMMON          RVS(3,801),RVB(3,801),RVV(3,801),RVT(3,801)       DCOMMN59
C                   (                (LMNT,LMBND1)               )      DCOMMN60
      DIMENSION       REVERB(9612),RV(3,801,3)                          DCOMMN61
C                          (LMRB),(LMNT,LMBND1,3)                       DCOMMN62
C                                                                       DCOMMN63
      EQUIVALENCE     (REVERB(1),RVS(1,1),RV(1,1,1))                    DCOMMN64
 END                                                                    DCOMMN65
DCOMN3  PROC                                                            DCOMMN66
      COMMON          XMIN(400),TMIN(400)                               DCOMMN67
      COMMON          RBX(400),RBT(400),RBTH(400),RBH(400)              DCOMMN68
C                 ( LMKS = MAX. NO. OF RAYS SORTED BY 'RAYSRT')         DCOMMN69
 END                                                                    DCOMMN70
```

APPENDIX G

OD 52258

PROCEDURES FOR DOP

```
DCOMN4  PROC                                                          DCOMMN71
      COMMON          FCSGAM(801)                                     DCOMMN72
C                       (LMBND1)                                      DCOMMN73
 END                                                                  DCOMMN74
DCOMN5  PROC                                                          DCOMMN75
      COMMON          X(800),T(800),THA(800),THB(800),R(800),DOP(800) DCOMMN76
      COMMON          COSTHA(800),COSTHB(800)                         DCOMMN77
      COMMON          OMT(800),COSOMT(800),SINOMT(800)                DCOMMN78
C                     (               LMKS2                    )      DCOMMN79
C                                                                     DCOMMN80
      COMMON          RBXA(400),RBTA(400),RBTHA(400),RBHA(400)        DCOMMN81
      DIMENSION       RBXB(400),RBTB(400),RBTHB(400),RBHB(400)        DCOMMN82
C                     (            (LMKS)             )               DCOMMN83
C                                                                     DCOMMN84
      EQUIVALENCE     (RBXB,DOP)                                      DCOMMN85
      EQUIVALENCE     (RBTB,T(400)),(RBTHB,THB(400)),(RBHB,R(400))    DCOMMN86
C                     (            (LMKS)                )            DCOMMN87
C                                                                     DCOMMN88
 END                                                                  DCOMMN89
DCOMN6  PROC                                                          DCOMMN90
      DATA            LMBAND/800/,LMBND1/801/,LMKS/400/,LMKS2/800/    DCOMMN91
      DATA            LMRB/9612/,LMTIM/400/,LMTRS/17/,LMSPRD/150/     DCOMMN92
C                                                                     DCOMMN93
C                     LMRB = 4 * LMNT * LMBND1                        DCOMMN94
C                     LMBND1 = LMBAND + 1                             DCOMMN95
C                     LMKS2 = LMKS * 2                                DCOMMN96
C                     LMNT = 3                                        DCOMMN97
 END                                                                  DCOMMN98
```

APPENDIX G

I N D E X PAGE 3

PROCEDURES FOR DOP

SYMBOL		= = = = = = = = = = = = = = =	REFERENCES	= = = = = = = = = = = = = = =		
ALPHC	-	27CO				
AR1	-	35CO	53IN			
BAND	-	5CO				
BNDOUT	-	5CO				
BR1	-	35CO	53IN			
BWIDTH	-	28CO				
BWINT	-	19CO				
CO	-	27CO				
CCKT	-	18CO				
CBAND	-	2CL	5CL			
CCOUNT	-	8CL				
CENTER	-	22CO	48EQ	54LG		
CFCNST	-	11CL	14CL	15CL		
CHCNST	-	16CL	17CL			
CINDAT	-	18CL	19CL			
CINDEF	-	21CL	22CL	23CL	24CL	
CINPUT	-	26CL	27CL	28CL	29CL	30CL
COSOMT	-	78CO				
COSTHA	-	77CO				
COSTHB	-	77CO				
CPRINT	-	33CL				
CSKSI	-	18CO				
CSPRED	-	34CL				
CTAPE	-	35CL				
CTBLKP	-	36CL				
CXCNST	-	37CL				
DO	-	26CO				
DBTTM	-	27CO				
DCOMN1	-	1				
DCOMN2	-	58				
DCOMN3	-	66				
DCOMN4	-	71				
DCOMN5	-	75				
DCOMN6	-	90				
DEGRAD	-	14CO				
DELDEP	-	36CO				
DELIND	-	36CO				
DELT	-	18CO				
DELT2	-	28CO				
DOP	-	76CO	85EQ			
DR	-	18CO				
END	-	22CO	54LG			
EXPS	-	19CO				
F0	-	43EQ				
F1	-	11CO				
F10	-	11CO				
F180	-	15CO				
F1E3	-	14CO				
F1MIN	-	15CO				
F2	-	11CO				
F20	-	11CO				
F3	-	15CO				
F4	-	11CO				
F90	-	15CO				
FACTOR	-	36CO				
FC03	-	19CO				
FCOVS	-	18CO				
FCSGAM	-	72CO				
FGAM	-	5CO				
FILTER	-	22CO	54LG			
FLOG10	-	14CO				
FNBAND	-	2CO				
FPT5	-	11CO				
FZ2	-	19CO				
FZRO	-	28CO	46EQ			
FZSQ	-	19CO				
GO	-	22CO	54LG			
HBAND	-	16CO				

APPENDIX G

PROCEDURES FOR DOP

APPENDIX G

APPENDIX G

I N D E X   PAGE 6

PROGRAM DOP

```
      INCLUDE DCOMN1
      INCLUDE DCOMN2
C
C        READ AND PROCESS INPUT, CARDS AND TAPE.
C
 9000 CALL IDENT
      IF (END) GO TO 20000
C
C        SET UP BAND LIMITS.
C
      CALL BCOMP
C
C        EXTRACT SURFACE AND/OR BOTTOM DATA FROM INPUT TAPE, AND
C        ARRANGE PROPERLY FOR DOP.  WRITE ON TEMPORARY FILE.
C
      CALL RBSORT
C
C        PROCESS TABLE OF REVERBERATION TIMES AS NECESSARY.
C
      CALL TCOMP
C
      NTMAX = 0
C
C        COMPUTE REVERBERATION FOR EACH TIME IN TABLE.
C
10000 NTMIN = NTMAX + K1
      NTMAX = MIN0(NTMAX + LMNT, NTIME)
      IF (SPREAD) NTMAX = NTMIN
C
C        ZERO OUT REVERBERATION TABLE.
C
      DO 10010 KT = 1, LMRB
      REVERB(KT) = 0.
10010 CONTINUE
C
C        COMPUTE A PAGE FULL--THREE TIMES OR ONE TIME WITH SPREADING.
C
      DO 15000 KT = NTMIN, NTMAX
      KTT = KT + K1 - NTMIN
C
C        COMPUTE BOUNDARY AND VOLUME REVERB. AND SPREAD AS REQUIRED.
C        COMPUTE TOTALS.
C
      IF (.NOT. NOTAPE) CALL RBCOMP
      IF (.NOT. NOVOLM) CALL RVCOMP
      IF (SPREAD) CALL RVSPRD
      CALL RTCOMP
15000 CONTINUE
C
C        PRINT REVERBERATION DATA.
      CALL RVPRNT
C
C
      IF (NTMAX .LT. NTIME) GO TO 10000
      GO TO 9000
C
C        REWIND TAPES AS REQUIRED, AND EXIT.
C
20000 IF (NOTAPE) GO TO 20010
      REWIND AR1
      REWIND BR1
20010 IF (PLOT) REWIND IPLT
30000 STOP
      END
```

APPENDIX G

I N D E X

PROGRAM DOP

SYMBOL		= = = = = = = = = = = = = = = = REFERENCES					
9000	-	6*	55				
10000	-	26*	54				
10010	-	32	34*				
15000	-	38	48*				
20000	-	7	59*				
20010	-	59	62*				
30000	-	63*					
AR1	-	60					
BCOMP	-	11					
BR1	-	61					
DCOMN1	-	1					
DCOMN2	-	2					
END	-	7					
IDENT	-	6					
IPLT	-	62					
K1	-	26	39				
KT	-	32=	33	36=	39		
KTT	-	39=					
LMNT	-	27					
LMRB	-	32					
MINO	-	27					
NOTAPE	-	44	59				
NOVOLM	-	45					
NTIME	-	27	54				
NTMAX	-	22=	26	27=	28=	38	54
NTMIN	-	26=	28	38	39		
PLOT	-	62					
RBCOMP	-	44					
RBSORT	-	16					
REVERB	-	33=					
RTCOMP	-	47					
RVCOMP	-	45					
RVPRNT	-	51					
RVSPRD	-	46					
SPREAD	-	28	46				
STOP	-	63					
TCOMP	-	20					
1MAIN	-	0					

APPENDIX G

```
      SUBROUTINE IDENT
C
      INCLUDE DCOMN1
      COMMON ID(12),I,J,K,L,M
      LOGICAL OUTFLG
      DATA INPFLG/0/,OUTFLG/.TRUE./
C
C         WRITE HEADING ON INPUT DATA PAGE.
C
      WRITE (IPRT,501) HED
  501 FORMAT (8A6,A4,16H-- INPUT DATA --/)
C
C         READ NEXT VARIABLE NAME AND RELATED DATA FROM DATA CARD(S).
C
  600 CALL INPUT(NNAMES,IDC,INPFLG,OUTFLG)
      IF (GO) GO TO 700
      IF (END) GO TO 900
      IF (INPFLG .GT. 1) END = .TRUE.
      GO TO 600
  700 GO = .FALSE.
      IF (NOTAPE) GO TO 2000
C
C         READ FIRST HEADER RECORD FROM INPUT TAPE.
C
      READ (AR1) (ID(I), I = 1, 6)
      IF (ID(6) .NE. 0) GO TO 1000
C
      END = .TRUE.
  900 WRITE (IPRT,901)
  901 FORMAT (69X,18H*** END OF RUN ***)
      GO TO 30000
C
C         READ SECOND HEADER RECORD.  PRESERVE ANY DATA FROM INPUT
C         TAPE WHICH HAS NOT BEEN READ IN FROM CARDS.
C
 1000 READ (AR1) (ID(I), I = 6, 12)
      DO 1500 I = 1, 12
      J = I
      IF (J .GT. K2) J = J - K1
      IF (NAMCNT(J) .EQ. 0) GO TO 1200
      IF (J .NE. 11) GO TO 1500
      KSI = KSI * DEGRAD
      GO TO 1500
 1200 IDATA(I) = ID(I)
 1500 CONTINUE
C
C         SCALE INPUT DATA AND PRECOMPUTE RELATED QUANTITIES.
C
 2000 OMEGA = OMEGAD * DEGRAD
      DELT = DELT2/F2
      KSID = KSI/DEGRAD
      BWINT = BWIDTH
      IF (.NOT. KTSBND) BWINT = BWINT/F1E3
      LOGMVI = LOGMV - LOG4PI
      CSKSI = COS(KSI)
      SNKSI = SIN(KSI)
      DR = DELT2 * CO/F4
      COKT = CO * TDKT
      FCOVS = COKT/VS
      FCO3 = CO**3 * TWOPI
      IF (FZRO .LE. 0.) FZRO = F1
      FZSQ = FZRO**2
      FZ2 = FZRO * F2
      EXPS = EXP(S/F10 * FLOG10)
      IF (THTMAX .EQ. 0.) THTMAX = F90
      IF (THTMAX .GT. F180) THTMAX = F180
C
      NSPRN1 = K1
      SPREAD = SPREAD .OR. (NSSPRD.NE.0) .OR. (NBSPRD.NE.0) .OR.
     1    (NVSPRD.NE.0)
```

APPENDIX G

SUBROUTINE IDENT

```
      IF (.NOT. SPREAD) GO TO 5010                                     IDENT 71
C                                                                      IDENT 72
C        GENERATE ANY REQUIRED MISSING SPREADING TABLES.               IDENT 73
C                                                                      IDENT 74
      IF ((NSSPRD .EQ. 0) .OR. (NDSPRD .EQ.0) .OR. (NVSPRD .EQ. 0))    IDENT 75
     1    CALL SPRCMP                                                  IDENT 76
      NSPRN1 = MAX0(NSSPRD,NDSPRD,NVSPRD)                              IDENT 77
C                                                                      IDENT 78
 5010 NSPRN = NSPRN1 - K1                                              IDENT 79
      NSPR1 = NSPRN + NSPRN1                                           IDENT 80
      IF (NOPRNT) GO TO 7010                                           IDENT 81
C                                                                      IDENT 82
C        SET UP APPROPRIATE PAGE HEADINGS.                             IDENT 83
C                                                                      IDENT 84
      K = 0                                                            IDENT 85
      J = K5                                                           IDENT 86
      IF (.NOT.SPREAD) GO TO 6020                                      IDENT 87
      J = 0                                                            IDENT 88
      IF (TVG) K = K10                                                 IDENT 89
      IF (FILTER) K = K + K5                                           IDENT 90
 6020 M = K1                                                           IDENT 91
      L = K1                                                           IDENT 92
      IF (KTSBND) L = 0                                                IDENT 93
      IF (.NOT. TOTALS) GO TO 6030                                     IDENT 94
      NTOT3(1) = HEADS(L+36)                                           IDENT 95
      L = K3                                                           IDENT 96
      M = K3                                                           IDENT 97
C                                                                      IDENT 98
 6030 L = L + K5                                                       IDENT 99
      DO 6040 I = K1, K5                                               IDENT100
      L = L + K1                                                       IDENT101
      NBAND(I) = HEADS(L)                                              IDENT102
      J = J + K1                                                       IDENT103
      NSPRD(I) = HEADS(J+10)                                           IDENT104
      K = K + K1                                                       IDENT105
      NSPRW(I) = HEADS(K+15)                                           IDENT106
 6040 CONTINUE                                                         IDENT107
      NFROM(1) = HEADS(M+37)                                           IDENT108
      NFROM(2) = HEADS(M+38)                                           IDENT109
 7010 IF (.NOT. NOTAPE) REWIND DR1                                     IDENT110
30000 RETURN                                                           IDENT111
      END                                                              IDENT112
```

SUBROUTINE IDENT

SYMBOL		REFERENCES									
501	-	10WR	11*								
600	-	15*	19								
700	-	16	20*								
900	-	17	29*								
901	-	29WR	30*								
1000	-	26	36*								
1200	-	40	44*								
1500	-	37	41	43	45*						
2000	-	21	49*								
5010	-	71	79*								
6020	-	87	91*								
6030	-	94	99*								
6040	-	100	107*								
7010	-	81	110*								
30000	-	31	111*								
AR1	-	25RD	36RD								
BR1	-	110									
BWIDTH	-	52									
BWINT	-	52=	53=								
CO	-	57	58	60							
COKT	-	58=	59								
COS	-	55									
CSKS1	-	55=									
DCOMN1	-	3									
DEGRAD	-	42	49	51							
DELT	-	50=									
DELT2	-	50	57								
DR	-	57=									
END	-	17	18=	28=							
EXP	-	64									
EXPS	-	64=									
F1	-	61									
F10	-	64									
F180	-	66									
F1E3	-	53									
F2	-	50	63								
F4	-	57									
F90	-	65									
FCO3	-	60=									
FCOVS	-	59=									
FILTER	-	90									
FLOG10	-	64									
FZ2	-	63=									
FZRO	-	61=	62	63							
FZSQ	-	62=									
GO	-	16	20=								
HBAND	-	102=									
HEADS	-	95	102	104	106	108	109				
HED	-	10WR									
HFROM	-	108=	109=								
HSPRD	-	104=									
HSPRW	-	106=									
HTOT3	-	95=									
I	-	4CO	25RD	36RD	37=	38	44	100=	102	104	106
ID	-	4CO	25RD	26	36RD	44					
IDATA	-	44=									
IDC	-	15AG									
IDENT	-	1									
INPFLG	-	6DA	15AG	18							
INPUT	-	15									
IPRT	-	10WR	29WR								
J	-	4CO	38=	39=	40	41	86=	88=	103=	104	
K	-	4CO	85=	89=	90=	105=	106				
K1	-	39	68	79	91	92	100	101	103	105	
K10	-	89									
K2	-	39									
K3	-	96	97								
K5	-	86	90	99	100						

APPENDIX G

I N D E X | SUBROUTINE IDENT | PAGE 11

-+-

APPENDIX G

```
      SUBROUTINE BCOMP
C
      INCLUDE BCOMM1
      COMMON B,BND(3),CNT(3),D,E,F,I,L,Q,
C
      BND(3) = COS(THTMAX * DEGRAD) * VS
      IF (THTMAX .EQ. F90) BND(3) = 0.
      IF (THTMAX .EQ. F180) BND(3) = -VS
      IF (KTSBND) GO TO 1000
C
C        OUTPUT BAND LIMITS IN KILOHERZ.
C
      Q = (COKT + VS)/(COKT - VS)
      BND(1) = FZRO * Q - FZRO
      B = FZRO
      BND(2) = FZRO - FZRO/Q
      D = FZRO
      BND(3) = (COKT + BND(3))/(COKT - BND(3)) * FZRO - FZRO
      GO TO 2000
C
C        OUTPUT BAND LIMITS IN KNOTS OF DOPPLER.
C
 1000 BND(1) = VS
      B = 0.
      BND(2) = VS
      D = VS
C
 2000 E = 0.
      IF (CENTER) E = BWINT * FPT5
      B = B + E
      IF (.NOT. RELBND) D = 0.
C
C        COMPUTE NO. OF BANDS WITH POSITIVE AND NEGATIVE DOPPLER,
C        ALSO BAND LIMIT DUE TO THETA MAX.
C
      DO 2500 I = 1, 3
      CNT(I) = ABS(AINT(BND(I)/BWINT))
      IF (CNT(I) * BWINT + E .LT. ABS(BND(I))) CNT(I) = CNT(I) + F1
      IF (CENTER) CNT(I) = CNT(I) + FPT5
 2500 CONTINUE
      CNT(3) = SIGN(CNT(3),BND(3))
      IF (BND(3) .GT. 0.) CNT(3) = CNT(3) - F1
C
C        MBAND = THEORETICAL NUMBER OF UNSPREAD BANDS.
C        NBAND - NSPRN = NO. OF UNSPREAD BANDS TO BE COMPUTED.
C        MNBAND = NO. OF BANDS TO BE OUTPUT, INCLUDING SPREADING.
C        NPSTRT = FIRST BAND TO BE PRINTED.
C
      FNBAND = CNT(1) + CNT(2)
      MBAND = FNBAND
      NBAND = MIN0(LMBAND,MBAND + NSPRN,IFIX(CNT(1) - CNT(3)) + NSPRN)
      MNBND = MIN0(LMBAND, NBAND + NSPRN)
      L = IFIX(CNT(1)) + NSPRN
      F = L
      IF (.NOT. CENTER) L = L - K1
      NPSTRT = MOD(L, IPEVRY) + K1
C
C        COMPUTE VALUES OF BAND LIMITS FOR COMPUTATION AND OUTPUT.
C
      L = MNBND + K1
      DO 3000 I = 1, L
      BAND(I) = BWINT * F + B
      BNDOUT(I) = BAND(I) - D
      IF (KTSBND) BAND(I) = FZRO * (COKT + BAND(I))/
     1                      (COKT - BAND(I))
      IF (FILTER) RRFS(I-1) = RRF((BAND(I-1)+BAND(I))*FPT5)
      F = F - F1
 3000 CONTINUE
C
C        RESET FLAG FOR VOLUME REVERB. PATTERN COMP. (NON-TURNING).
```

APPENDIX G

```
      C                                                                  BCOMP 71
            VPTTRN = .FALSE.                                             BCOMP 72
      C                                                                  BCOMP 73
            RETURN                                                       BCOMP 74
            END                                                          BCOMP 75
```

APPENDIX G

```
I N D E X                                                                  PAGE  14
                                   SUBROUTINE BCOMP

SYMBOL      - - - - - - - - - - - - - - -    REFERENCES    - - - - - - - - - - - - - - -

1000     -     9     23*
2000     -    19     28*
2500     -    36     40*
3000     -    61     68*
ABS      -    37     38
AINT     -    37
B        -   4CO     15=     24=     30=     62
BAND     -   62=     63      64=     65      66
BCOMP    -     1
BND      -   4CO      6=      7=      8=     14=     16=     18=     23=     25=     37
              38     41      42
BNDOUT   -   63=
BWINT    -   29      37      38      62
COKT     -   13      18      64      65
CENTER   -   29      39      55
CNT      -   4CO     37=     38=     39=     41=     42=     49      51      53
COS      -     6
D        -   4CO     17=     26=     31=     63
DCOMN1   -     3
DEGRAD   -     6
E        -   4CO     28=     29=     30      38
F        -   4CO     54=     62      67=
F1       -   38      42      67
F180     -     8
F90      -     7
FILTER   -   66
FNBAND   -   49=     50
FPTS     -   29      39      66
FZRO     -   14      15      16      17      18      64
I        -   4CO     36=     37      38      39      61=     62      63      64      65
              66
IFIX     -   51      53
IPEVRY   -   56
K1       -   55      56      60
KTSBND   -     9     64
L        -   4CO     53=     54      55=     56      60=     61
LMBAND   -   51      52
MBAND    -   50=     51
MIND     -   51      52
MNBND    -   52=     60
MOD      -   56
NBAND    -   51=     52
NPSTRT   -   56=
NSPRN    -   51      52      53
Q        -   4CO     13=     14      16
RELBND   -   31
RETURN   -   74
RRF      -   66
RRFS     -   66=
SIGN     -   41
TNTMAX   -     6      7       8
VPTTRN   -   72=
VS       -     6      8      13      23      25      26
```

APPENDIX G

```
      SUBROUTINE  RBSORT
C
      INCLUDE DCOMN1
      INCLUDE DCOMN3
      COMMON I,J,K,L,M,N,NA,NB,NC,ND,A,D,TEST,WRFLG
      LOGICAL WRFLG
C
      NSURF = 0
      NBTM = 0
      IF (NOTAPE) GO TO 16
C
      TEST = DBTTM * SHIFT
      L = 0
C
C         READ DEPTH ID. RECORD AND TEST FOR SURFACE.
C
    1 WRFLG = .FALSE.
      READ (AR1) I, D, N
      IF (I .EQ. 0) GO TO 15
      IF (D .NE. 0.) GO TO 2
      IF (NOSURF) GO TO 7
      NSURF = N
      GO TO 5
C
C         TEST FOR BOTTOM DEPTH.
C
    2 IF (ABS(D - DBTTM) .GT. TEST) GO TO 7
      IF (NOBTTM) GO TO 7
      NBTM = N
C
C         READ DATA FOR NEXT PATH.  SKIP PATHS NOT AT (DESIRED) SURFACE
C         OR BOTTOM.
C
    5 WRFLG = .TRUE.
    7 DO 14 I = 1, N
      READ (AR1) J,(RBTH(K),RBX(K),RBH(K),RBT(K), K = 1, J)
      IF (.NOT. WRFLG) GO TO 14
      DO 9 K = 1, J
      RBH(K) = RBH(K) - S
    9 CONTINUE
C
C         PUT DATA IN ASCENDING ORDER OF X, IF NECESSARY.
C
      IF (RBX(2) .GT. RBX(1)) GO TO 12
      NA = J/2
      NB = -LMKS
      DO 11 K = 1, 4
      NB = NB + LMKS
      NC = NB + J
      DO 10 M = 1, NA
      ND = NB + M
      A = RBX(ND)
      RBX(ND) = RBX(NC)
      RBX(NC) = A
      NC = NC - 1
   10 CONTINUE
   11 CONTINUE
C
C         SAVE MINIMUM X AND T FOR EACH PATH, IF TIMCMP OPTION.
C
      IF (.NOT. TIMCMP) GO TO 13
   12 L = L + 1
      XMIN(L) = RBX(1)
      TMIN(L) = RBT(1)
C
C         WRITE DATA ON INTERMEDIATE TAPE.
C
   13 WRITE (BR1) J,(RBX(K),RBT(K),RBTH(K),RBH(K), K = 1, J)
   14 CONTINUE
      GO TO 1
```

OD 52258

```
I N D E X                                                               PAGE  16
                              SUBROUTINE RBSORT

      C                                                                 RBSORT71
   15 REWIND BR1                                                        RBSORT72
   16 RETURN                                                            RBSORT73
      END                                                               RBSORT74
```

SUBROUTINE RBSORT

SYMBOL		* * * * * * * * * * * * * * * * REFERENCES * * * * * * * * * * * * * * * *						
1	-	17*	70					
2	-	20	27*					
5	-	23	34*					
7	-	21	27	28	35*			
9	-	38	40*					
10	-	50	56*					
11	-	47	57*					
12	-	44	62*					
13	-	61	68*					
14	-	35	37	69*				
15	-	19	72*					
16	-	10	73*					
A	-	5CO	52=	54				
ABS	-	27						
AR1	-	18RD	36RD					
BR1	-	68WR	72					
B	-	5CO	18RD	20	27			
BBTTM	-	12	27					
BCOMN1	-	3						
BCOMN3	-	4						
I	-	5CO	18RD	19	35=			
J	-	5CO	36RD	38	45	49	68WR	
K	-	5CO	36RD	38=	39	47=	68WR	
L	-	5CO	13=	62=	63	64		
LMKS	-	46	48					
M	-	5CO	50=	51				
N	-	5CO	18RD	22	29	35		
NA	-	5CO	45=	50				
NB	-	5CO	46=	48=	49	51		
NBTM	-	9=	29=					
NC	-	5CO	49=	53	54	55=		
ND	-	5CO	51=	52	53			
NOBTTM	-	28						
NOSURF	-	21						
NOTAPE	-	10						
NSURF	-	8=	22=					
RBN	-	36RD	39=	68WR				
RBSORT	-	1						
RBT	-	36RD	64	68WR				
RBTN	-	36RD	68WR					
RBX	-	36RD	44	52	53=	54=	63	68WR
RETURN	-	73						
S	-	39						
SHIFT	-	12						
TEST	-	5CO	12=	27				
TIMCMP	-	61						
TMIN	-	64=						
WRFLG	-	5CO	6LG	17=	34=	37		
XMIN	-	63=						

OD 52258


```
      SUBROUTINE  TCOMP                                                 TCOMP  1
C                                                                       TCOMP  2
      INCLUDE DCOMN1                                                    TCOMP  3
      INCLUDE DCOMN3                                                    TCOMP  4
      COMMON I,J,K,M,IBOUND,NMIN,NMAX,NTBL                              TCOMP  5
C                                                                       TCOMP  6
      IF (.NOT. TIMCMP) GO TO 10                                        TCOMP  7
C                                                                       TCOMP  8
C        TIME COMPUTATION IS DESIRED.  NOTE THAT CONSTANT CHECKING IS   TCOMP  9
C        DONE TO ENSURE THAT ALL ADDED TIMES LIE BETWEEN 1/2 DELTA T    TCOMP 10
C        AND PING INTERVAL. NO SUCH CHECK IS MADE ON INPUT TIMES.       TCOMP 11
C        ALSO LENGTH OF THE TIME ARRAY IS CHECKED TO AVOID OVER-FILLING.TCOMP 12
C                                                                       TCOMP 13
      IF (NTIME .EQ. LMTIM) GO TO 10                                    TCOMP 14
      NTIME = NTIME + K1                                                TCOMP 15
      IF (NTIME .EQ. LMTIM) GO TO 9                                     TCOMP 16
      TIME(NTIME) = DELT                                                TCOMP 17
      NTIME = NTIME + K1                                                TCOMP 18
      IF (NTIME .EQ. LMTIM) GO TO 9                                     TCOMP 19
      IF (NOTAPE) GO TO  6                                              TCOMP 20
C                                                                       TCOMP 21
C        COMPUTE TIMES ASSOCIATED WITH WAVE-FRONT ARRIVAL OVER EACH     TCOMP 22
C        COMBINATION OF TWO-WAY PATHS.                                  TCOMP 23
C                                                                       TCOMP 24
      NMAX = 0                                                          TCOMP 25
      DO 4 IBOUND = K1, K2                                              TCOMP 26
      IF (NBOUND(IBOUND) .EQ. 0) GO TO 4                                TCOMP 27
      NMIN = NMAX + K1                                                  TCOMP 28
      NMAX = NMAX + NBOUND(IBOUND)                                      TCOMP 29
      DO 3 I = NMIN, NMAX                                               TCOMP 30
      READ (BR1) NTBL,(RBX(K),RBT(K),RBTH(K),RBH(K), K = 1, NTBL)       TCOMP 31
C                                                                       TCOMP 32
      DO 2 J = I, NMAX                                                  TCOMP 33
      TIME(NTIME) = (TMIN(J) + TABLKP(XMIN(J),RBX,RBT,1,NTBL)) + FPT5   TCOMP 34
      IF (ITABLE .EQ. 0) GO TO 2                                        TCOMP 35
      IF (TIME(NTIME) .GE. PING) GO TO 2                                TCOMP 36
      M = NTIME                                                         TCOMP 37
      IF (TIME(NTIME) .LE. DELT) GO TO  1                               TCOMP 38
      NTIME = NTIME + K1                                                TCOMP 39
      IF (NTIME .EQ. LMTIM) GO TO 9                                     TCOMP 40
      TIME(NTIME) = TIME(M) - DELT                                      TCOMP 41
      IF (TIME(NTIME) .GT. DELT) NTIME = NTIME + K1                     TCOMP 42
      IF (NTIME .EQ. LMTIM) GO TO 9                                     TCOMP 43
    1 TIME(NTIME) = TIME(M) + DELT                                      TCOMP 44
      IF (TIME(NTIME) .LT. PING) NTIME = NTIME + K1                     TCOMP 45
      IF (NTIME .EQ. LMTIM) GO TO 9                                     TCOMP 46
    2 CONTINUE                                                          TCOMP 47
    3 CONTINUE                                                          TCOMP 48
    4 CONTINUE                                                          TCOMP 49
      REWIND BR1                                                        TCOMP 50
C                                                                       TCOMP 51
C        ADD PRESET TABLE OF TIMES.                                     TCOMP 52
C                                                                       TCOMP 53
    6 DO 7 I = K1, LMTRS                                                TCOMP 54
      IF (TRS(I) .GE. PING) GO TO 9                                     TCOMP 55
      TIME(NTIME) = TRS(I)                                              TCOMP 56
      IF (TIME(NTIME) .GT. DELT) NTIME = NTIME + K1                     TCOMP 57
      IF (NTIME .EQ. LMTIM) GO TO 9                                     TCOMP 58
    7 CONTINUE                                                          TCOMP 59
C                                                                       TCOMP 60
C        CONTINUE WITH EVERY 1/2 SECOND TO PING.  ADD PING INTERVAL.    TCOMP 61
C                                                                       TCOMP 62
    8 TIME(NTIME) = TIME(NTIME-1) + FPT5                                TCOMP 63
      IF (TIME(NTIME) .GE. PING) GO TO 9                                TCOMP 64
      NTIME = NTIME + K1                                                TCOMP 65
      IF (NTIME .NE. LMTIM) GO TO 8                                     TCOMP 66
C                                                                       TCOMP 67
    9 TIME(NTIME) = PING                                                TCOMP 68
C                                                                       TCOMP 69
C        SORT TIME ARRAY INTO ASCENDING ORDER, ELIMINATING DUPLICATES.  TCOMP 70
```

APPENDIX G

SUBROUTINE TCOMP

```
      C                                                                 TCOMP 71
   10 CALL SORT(TIME,NTIME)                                             TCOMP 72
      C                                                                 TCOMP 73
      C     INITIATE PAGE COUNTER AND COMPUTE TOTAL NUMBER OF PAGES OF  TCOMP 74
      C     OUTPUT FOR THIS FILE OF INPUT DATA.                         TCOMP 75
      C                                                                 TCOMP 76
         PAGE = 0                                                       TCOMP 77
         NMAX = (NTIME - K1)/K3 + K1                                    TCOMP 78
         IF (SPREAD) NMAX = NTIME                                       TCOMP 79
         NPAGE = (((NNBND - NPSTRT)/IPEVRY)/K40 + K1) * NMAX            TCOMP 80
         IF (TOTALS) NPAGE = (NTIME - K1)/K40 + K1                      TCOMP 81
      C                                                                 TCOMP 82
         RETURN                                                         TCOMP 83
         END                                                            TCOMP 84
```

APPENDIX G

I N D E X PAGE 20

SUBROUTINE TCOMP

SYMBOL		= = = = = = = = = = = = = = = REFERENCES = = = = = = = = = = = = = = =									
1	-	38	44*								
2	-	33	35	36	47*						
3	-	30	48*								
4	-	26	27	49*							
6	-	20	54*								
7	-	54	59*								
8	-	63*	66								
9	-	16	19	40	43	46	55	58	64	68*	
10	-	7	14	72*							
BR1	-	31RD	50								
DCOMN1	-	3									
DCOMN3	-	4									
DELT	-	17	38	41	42	44	57				
FPTS	-	34	63								
I	-	5CO	30=	33	54=	55	56				
IBOUND	-	5CO	26=	27	29						
IPEVRY	-	80									
ITABLE	-	35									
J	-	5CO	33=	34							
K	-	5CO	31RD								
K1	-	15	18	26	28	39	42	45	54	57	65
		78	80	81							
K2	-	26									
K3	-	78									
K40	-	80	81								
LMTIM	-	14	16	19	40	43	46	58	66		
LMTRS	-	54									
M	-	5CO	37=	41	44						
MNBND	-	80									
NBOUND	-	27	29								
NMAX	-	5CO	25=	28	29=	30	33	78=	79=	80	
NMIN	-	5CO	28=	30							
NOTAPE	-	20									
NPAGE	-	80=	81=								
NPSTRT	-	80									
NTBL	-	5CO	31RD	34							
NTIME	-	14	15=	16	17	18=	19	34	36	37	38
		39=	40	41	42=	43	44	45=	46	56	57=
		58	63	64	65=	66	68	72AG	78	79	81
PAGE	-	77=									
PING	-	36	45	55	64	68					
RPM	-	31RD									
RBT	-	31RD	34								
RBTH	-	31RD									
RBX	-	31RD	34								
RETURN	-	83									
SORT	-	72									
SPREAD	-	79									
TABLKP	-	34									
TCOMP	-	1									
TIMCMP	-	7									
TIME	-	17=	34=	36	38	41=	42	44=	45	56=	57
		63=	64	68=	72AG						
TMIN	-	34									
TOTALS	-	81									
TRS	-	55	56								
XMIN	-	34									

APPENDIX G

```
      SUBROUTINE RBCOMP                                                 RBCOMP 1
C                                                                       RBCOMP 2
      INCLUDE DCOMN1                                                    RBCOMP 3
      INCLUDE DCOMN2                                                    RBCOMP 4
      INCLUDE DCOMN5                                                    RBCOMP 5
      COMMON DD(2),XD(2),TD(2),THAD(2),THBD(2),RD(2)                    RBCOMP 6
      COMMON CSTHAD(2),CSTHBD(2),OMTD(2),CSOMTD(2),SNOMTD(2)            RBCOMP 7
      COMMON XTMU,OMTTMU,COMTMU,SOMTMU,CTHTMU(2)                        RBCOMP 8
      COMMON RT(2),THAT(2),THBT(2)                                      RBCOMP 9
      COMMON ARG,FA,QA,QB,QC,QAB,QAC,QBC,QABC,QAABC,QBABC               RBCOMP10
      COMMON CSALFA,CSALFB,CSBETA,CSBETB,CSGAMA,CSGAMB                  RBCOMP11
      COMMON CALFAP,CALFBP,CGAMAP,CGAMBP,COSA,COSB                      RBCOMP12
      COMMON PHI,PHMOMT,DELR,TWMAX,TWMIN,TTMAX,A,DOPSKP                 RBCOMP13
      COMMON X1(2),X2(2),X3(2),X4(2),Y1(2),Y2(2),Y3(2),Y4(2)            RBCOMP14
      COMMON I,II,IJ,JA,JB,JBOUND,JC,JD,JDD,JDELT,JDMAX,JDMIN,JE        RBCOMP15
      COMMON JMAX,JUMP,KBAND,KTIM,L,LEAP,LL,MLMU(2),MM,MN,MX,NA,NB,NEXT RBCOMP16
      EQUIVALENCE (MLMU(1),ML),(MLMU(2),MU),(DOP,IDOP),(MNMX,MN)        RBCOMP17
      EQUIVALENCE (QCOS,QAABC),(QSIN,QBABC),(QCOSM,QAB)                 RBCOMP18
      DIMENSION IDOP(1),MNMX(2)                                         RBCOMP19
      LOGICAL TTMAX,DOPSKP                                              RBCOMP20
C                                                                       RBCOMP21
C         BOUNDARY LOOP--                                               RBCOMP22
C         ONCE THROUGH EACH FOR SURFACE AND BOTTOM                      RBCOMP23
C                                                                       RBCOMP24
      DO 9010 JBOUND = 1, 2                                             RBCOMP25
      JMAX = NBOUND(JBOUND)                                             RBCOMP26
      IF (JMAX .EQ. 0) GO TO 9010                                       RBCOMP27
C                                                                       RBCOMP28
C         PATH LOOP--                                                   RBCOMP29
C         ONCE THROUGH FOR EACH SURFACE REFLECTED PATH AND EACH BOTTOM  RBCOMP30
C         REFLECTED PATH ON THE INTERMEDIATE TAPE.  (PATH A)            RBCOMP31
C                                                                       RBCOMP32
      DO 8010 JA = 1, JMAX                                              RBCOMP33
C                                                                       RBCOMP34
C         POSITION TAPE AT JA'TH RECORD.                                RBCOMP35
C                                                                       RBCOMP36
      IF (JA .EQ. 1) GO TO 1020                                         RBCOMP37
      DO 1010 I = JA,JMAX                                               RBCOMP38
      BACKSPACE BR1                                                     RBCOMP39
 1010 CONTINUE                                                          RBCOMP40
C                                                                       RBCOMP41
 1020 READ (BR1) NA,(RBXA(I),RBTA(I),RBTHA(I),RBHA(I), I = 1, NA)       RBCOMP42
      BACKSPACE BR1                                                     RBCOMP43
C                                                                       RBCOMP44
C         PATH COMBINATION LOOP--                                       RBCOMP45
C         ONCE THROUGH FOR EACH COMBINATION OF PATHS TO SURFACE         RBCOMP46
C         OR TO BOTTOM.  (PATH B)                                       RBCOMP47
C                                                                       RBCOMP48
      DO 7080 JB = JA, JMAX                                             RBCOMP49
      READ (BR1) NB,(RBXB(I),RBTB(I),RBTHB(I),RBHB(I), I = 1, NB)       RBCOMP50
C                                                                       RBCOMP51
C         DEFINE THE AREA COMMON TO THE TWO PATHS AND ALSO TO THE       RBCOMP52
C         CURRENT TIME INTERVAL, IF ANY SUCH AREA EXISTS.               RBCOMP53
C                                                                       RBCOMP54
      TWMIN = AMAX1(TIME(KT)-DELT,RBTA(1),RBTB(1))                      RBCOMP55
      TWMAX = AMIN1(TIME(KT)+DELT,RBTA(NA),RBTB(NB))                    RBCOMP56
      IF (TWMAX .LE. TWMIN) GO TO 7080                                  RBCOMP57
C                                                                       RBCOMP58
      DO 1030 JC = 1, NA                                                RBCOMP59
      X(JC) = RBXA(JC)                                                  RBCOMP60
 1030 CONTINUE                                                          RBCOMP61
      JC = NA                                                           RBCOMP62
C                                                                       RBCOMP63
      DO 1040 I = 1, NA                                                 RBCOMP64
      JC = JC + K1                                                      RBCOMP65
      X(JC) = RBXB(I)                                                   RBCOMP66
 1040 CONTINUE                                                          RBCOMP67
      CALL SORT(X, JC)                                                  RBCOMP68
C                                                                       RBCOMP69
C         EVALUATE TABLES FOR OVERLAPPED AREA.                          RBCOMP70
```

OD 52258


```
      C
            JD = 0
            II = K1
            IJ = K1
            DO 1050 I = 1, JC
            T(JD+1) = TABLKP(X(I),RBXA,RBTA,II,NA)
            IF (ITABLE .EQ. 0) GO TO 1050
            II = ITABLE
            T(JD+1) = (T(JD+1) + TABLKP(X(I),RBXB,RBTB,IJ,NB)) * FPT5
            IF (ITABLE .EQ. 0) GO TO 1050
            IJ = ITABLE
            JD = JD + K1
            X(JD) = X(I)
            THB(JD) = AINTRP(RBTHB) * DEGRAD
            R(JD) = AINTRP(RBMB)
            THA(JD) = TABLKP(X(I),RBXA,RBTHA,II,NA) * DEGRAD
            R(JD) = EXP(FLOG10 * (AINTRP(RBMA) + R(JD))/F20)
            COSTHA(JD) = COS(THA(JD))
            COSTHB(JD) = COS(THB(JD))
            OMT(JD) = AMOD(OMEGA * T(JD), TWOPI)
            IF (ABS(OMT(JD)) .GT. PI) OMT(JD) = OMT(JD) - SIGN(TWOPI, OMT(JD))
            COSOMT(JD) = COS(OMT(JD))
            SINOMT(JD) = SIGN(SQRT(F1-COSOMT(JD)**2),OMT(JD))
 1050 CONTINUE
      C
      C     TRANSMIT-RECEIVE LOOP--
      C     JC = 1, TRANSMIT PATH A, RECEIVE PATH B
      C     JC = 2, TRANSMIT PATH B, RECEIVE PATH A
      C
            DO 7070 JC = 1,2
      C
      C     FORE - AFT LOOP--
      C     JE = 1, FORWARD HEMISPHERE, OR RECEIVE FREQUENCY GREATER
      C     THAN TRANSMIT FREQUENCY.
      C     JE = 2, AFTER HEMISPHERE, OR RECEIVE FREQ. LESS THAN XMIT FREQ.
      C
            DO 7050 JE = 1, 2
            LL = (JC - K1) * LMKS2
            MM = LMKS2 - LL
            FA = K3 - JE * K2
            QCOS = FA
            QCOSM = FA
            DO 2020 I = 1, JD
            ML = LL + I
            MU = MM + I
            IF (OMEGA .EQ. FO) GO TO 2010
            QA = COSTHA(MU) * COSOMT(I) + COSTHA(ML)
            QB = COSTHA(MU) * SINOMT(I)
            QC = QB * COSTHA(ML)/FCOVS
            QAB = QA**2 + QB**2
            QCOS = (QB * QC + FA * QA * SQRT(QAB - QC**2))/QAB
            QSIN = (QB * QCOS - QC)/QA
            QCOSM = (QCOS * COSOMT(I) + QSIN * SINOMT(I))
 2010 DOP(I) = FZRO * (COKT + VS * COSTHA(MU) * QCOSM)
     1                 /(COKT - VS * COSTHA(ML) * QCOS)
 2020 CONTINUE
            MN = K1
            DOPSKP = .FALSE.
            SQRTMU = 0.
      C
      C     LOOP TO PROCESS CONSECUTIVE STRINGS OF DATA WHICH ARE
      C     MONOTONIC IN "DOP"
      C
 3010 JDELT = ISIGN(1,IDOP(MN) - IDOP(MN+1))
            ML = K1 + (JDELT + K1)/K2
            MU = K3 - ML
            IJ = JD - K1
            DO 3020 I = MN, IJ
            IF (JDELT .NE. ISIGN(1, IDOP(I) - IDOP(I+1))) GO TO 3030
 3020 CONTINUE
```

SUBROUTINE RBCOMP

```
      I = JD
C
 3030 MX = I
      JDMIN = MNMX(ML)
      JDMAX = MNMX(MU)
C
      DD(1) = DOP(JDMAX)
      IF (JE .EQ. K2) DD(1) = FZRO
      JDD = JDMAX
      ASSIGN 3050 TO NEXT
      DO 3040 KBAND = NSPRM1, NBAND
      IF (DD(1) .GT. BAND(KBAND+1)) GO TO (3070, 3090), JE
 3040 CONTINUE
      GO TO 7030
C
C         LOOP FOR ALL DESIRED BANDS AND FOR ALL DATA AT CONSTANT DOPPLER
C         WITHIN A BAND, IF BANDWIDTH IS GREATER THAN DATA SPACING.
C         IN THE FOLLOWING TABLES, THE SUBSCRIPTS 1 AND 2 REFER TO
C         DATA POINTS OF THE LOWEST AND HIGHEST FREQUENCY--SUBSCRIPTS
C         ML AND MU REFER TO POINTS OF THE LOWEST AND HIGHEST TIMES.
C
 3050 ASSIGN 4020 TO NEXT
C
 3060 DD(2) = DD(1)
      XD(2) = XD(1)
      TD(2) = TD(1)
      THAD(2) = THAD(1)
      THBD(2) = THBD(1)
      RD(2) = RD(1)
      CSTHAD(2) = CSTHAD(1)
      CSTHBD(2) = CSTHBD(1)
      OMTD(2) = OMTD(1)
      CSOMTD(2) = CSOMTD(1)
      SNOMTD(2) = SNOMTD(1)
C
      JDD = JDD + JDELT
      IF (DOP(JDD) .LT. BAND(KBAND+1)) GO TO 3080
      DD(1) = DOP(JDD)
 3070 XD(1) = X(JDD)
      TD(1) = T(JDD)
      THAD(1) = THA(JDD)
      THBD(1) = THB(JDD)
      RD(1) = R(JDD)
      CSTHAD(1) = COSTHA(JDD)
      CSTHBD(1) = COSTHB(JDD)
      OMTD(1) = OMT(JDD)
      CSOMTD(1) = COSOMT(JDD)
      SNOMTD(1) = SINOMT(JDD)
      GO TO 4010
C
 3080 JDD = JDD - JDELT
      DD(1) = BAND(KBAND+1)
 3090 XD(1) = TABLKP(DD(1),DOP,X,MN,MX)
      DOPSKP = ITABLE .EQ. 0
      IF (DOPSKP) GO TO NEXT
      TD(1)= AINTRP(T)
      THAD(1) = AINTRP(THA)
      THBD(1) = AINTRP(THB)
      RD(1) = AINTRP(R)
      CSTHAD(1) = AINTRP(COSTHA)
      CSTHBD(1) = AINTRP(COSTHB)
      OMTD(1) = AINTRP(OMT)
      CSOMTD(1) = AINTRP(COSOMT)
      SNOMTD(1) = AINTRP(SINOMT)
C
 4010 GO TO NEXT
C
 4020 ASSIGN 6010 TO LEAP
      KTIM = K1
      TTMAX = .FALSE.
```

APPENDIX G

OD 52258

SUBROUTINE RBCOMP

```
C
      IF (DOPSKP) GO TO 4040
      IF (MLMU(JE) .EQ. K1) GO TO 4030
      IF (TWMIN .LT. TD(MU)) GO TO 4040
      GO TO 7020
C
 4030 IF (TWMAX .LT. TD(ML)) GO TO 7020
      IF (TWMIN .LT. TD(ML)) GO TO 4080
 4040 ARG = TWMIN
      GO TO 4120
C
C           LOOP FOR ALL TIMES AT CONSTANT DOPPLER
C
 4050 IF (X(KTIM) .LE. XTMU) GO TO 7010
      RT(ML) = RT(MU)
      THAT(ML) = THAT(MU)
      THBT(ML) = THBT(MU)
C
      ARG = AMIN1(T(KTIM), TWMAX)
      ASSIGN 4110 TO JUMP
      IF (ARG .LT. TWMAX) ASSIGN 4100 TO JUMP
      IF (DOPSKP) GO TO JUMP
      IF (MLMU(JE) .EQ. K2) GO TO 4070
      IF (XTMU .GE. XD(MU)) GO TO JUMP
 4060 IF (ARG .LT. TD(MU)) GO TO JUMP
      MM = MU
      GO TO 4090
C
 4070 IF (ARG .LT. TD(ML)) GO TO JUMP
      IF (XTMU .LT. XD(ML)) GO TO 4080
      IF (ARG .GE. TD(MU)) TTMAX = .TRUE.
      GO TO 4060
 4080 MM = ML
C
 4090 XTMU = XD(MM)
      RT(MU) = RD(MM)
      THAT(MU) = THAD(MM)
      THBT(MU) = THBD(MM)
      CTHTMU(1) = CSTHAD(MM)
      CTHTMU(2) = CSTHBD(MM)
      OMTTMU = OMTD(MM)
      COMTMU = CSOMTD(MM)
      SOMTMU = SNOMTD(MM)
      KTIM = KTIM - K1
      GO TO 4130
C
 4100 XTMU = X(KTIM)
      RT(MU) = R(KTIM)
      THAT(MU) = THA(KTIM)
      THBT(MU) = THB(KTIM)
      CTHTMU(1) = COSTHA(KTIM)
      CTHTMU(2) = COSTHB(KTIM)
      OMTTMU = OMT(KTIM)
      COMTMU = COSOMT(KTIM)
      SOMTMU = SINOMT(KTIM)
      GO TO 4130
C
 4110 TTMAX = .TRUE.
 4120 XTMU = TABLKP(ARG,T,X,KTIM,JD)
      RT(MU) = AINTRP(R)
      THAT(MU) = AINTRP(THA)
      THBT(MU) = AINTRP(THB)
      CTHTMU(1) = AINTRP(COSTHA)
      CTHTMU(2) = AINTRP(COSTHB)
      OMTTMU = AINTRP(OMT)
      COMTMU = AINTRP(COSOMT)
      SOMTMU = AINTRP(SINOMT)
C
 4130 LL = K3 - JC
C
```

APPENDIX G

INDEX

SUBROUTINE RBCOMP

```
C       LOOP FOR POINTS 3 AND 4--                                       RBCOM281
C       I = 1, COMPUTE FOR POINTS 1 AND 3                               RBCOM282
C       I = 2, COMPUTE FOR POINTS 2 AND 4                               RBCOM283
C                                                                       RBCOM284
      DO 5020 I = 1, 2                                                  RBCOM285
      L = K3 - I                                                        RBCOM286
      FA = FZRO * CTHTMU(LL)                                            RBCOM287
      QA = DD(L) * CTHTMU(JC) + FA * COMTMU                             RBCOM288
      QB = FA * SOMTMU                                                  RBCOM289
      QC = FCOVS * (DD(L) - FZRO)                                       RBCOM290
      QAB = QA**2 + QB**2                                               RBCOM291
      QAC = QA * QC                                                     RBCOM292
      QBC = QB * QC                                                     RBCOM293
      QABC = SQRT(AMAX1(F0, QAB - QC**2))                               RBCOM294
      QAABC = QA * QABC                                                 RBCOM295
      QBABC = QB * QABC                                                 RBCOM296
      II = K2 * (I - K1)                                                RBCOM297
C                                                                       RBCOM298
C       PORT - STARBOARD LOOP--                                         RBCOM299
C       L = 1, AREA IN FIRST QUADRANT (OMEGA = 0)                       RBCOM300
C       L = 2, AREA IN FOURTH QUADRANT (OMEGA = 0)                      RBCOM301
C                                                                       RBCOM302
      DO 5010 L = 1, 2                                                  RBCOM303
      IJ = L + II                                                       RBCOM304
      A = K3 - L * K2                                                   RBCOM305
      FA = XTMU/QAB                                                     RBCOM306
C                                                                       RBCOM307
      X1(IJ) = X3(IJ)                                                   RBCOM308
      X3(IJ) = FA * (QAC - A * QBABC)                                   RBCOM309
      Y1(IJ) = Y3(IJ)                                                   RBCOM310
      Y3(IJ) = FA * (QBC + A * QAABC)                                   RBCOM311
 5010 CONTINUE                                                          RBCOM312
 5020 CONTINUE                                                          RBCOM313
C                                                                       RBCOM314
      GO TO LEAP                                                        RBCOM315
 6010 ASSIGN 6020 TO LEAP                                               RBCOM316
      GO TO 7010                                                        RBCOM317
C                                                                       RBCOM318
 6020 CSGAMA = SIN((THAT(MU) + THAT(ML))/F2)                            RBCOM319
      CSGAMB = SIN((THBT(MU) + THBT(ML))/F2)                            RBCOM320
      COSA = SQRT(F1 - CSGAMA**2)                                       RBCOM321
      COSB = SQRT(F1 - CSGAMB**2)                                       RBCOM322
C                                                                       RBCOM323
C       PORT - STARBOARD LOOP--                                         RBCOM324
C       L = 1, AREA IN FIRST QUADRANT (OMEGA = 0)                       RBCOM325
C       L = 2, AREA IN FOURTH QUADRANT (OMEGA = 0)                      RBCOM326
C                                                                       RBCOM327
      DO 6030 L = 1,2                                                   RBCOM328
      A = ((X4(L)*Y2(L) + X3(L)*Y4(L) + X2(L)*Y1(L) + X1(L)*Y3(L)) -    RBCOM329
     1     (X2(L)*Y4(L) + X4(L)*Y3(L) + X1(L)*Y2(L) + X3(L)*Y1(L)))     RBCOM330
      A = ABS(A/(RT(MU) + RT(ML)))                                      RBCOM331
C                                                                       RBCOM332
      PHI = ATAN2(Y1(L) + Y2(L) + Y3(L) + Y4(L),                        RBCOM333
     1            X1(L) + X2(L) + X3(L) + X4(L))                        RBCOM334
      PHMOMT = PHI - OMTTMU                                             RBCOM335
C                                                                       RBCOM336
      CSALFA = COSA * COS(PHI)                                          RBCOM337
      CSBETA = COSA * SIN(PHI)                                          RBCOM338
      CSALFB = COSB * COS(PHMOMT)                                       RBCOM339
      CSBETB = COSB * SIN(PHMOMT)                                       RBCOM340
C                                                                       RBCOM341
      CALFAP = CSALFA * CSKSI + CSGAMA * SNKSI                          RBCOM342
      CGAMAP = CSGAMA * CSKSI - CSALFA * SNKSI                          RBCOM343
      CALFBP = CSALFB * CSKSI + CSGAMB * SNKSI                          RBCOM344
      CGAMBP = CSGAMB * CSKSI - CSALFB * SNKSI                          RBCOM345
C                                                                       RBCOM346
      DELR = A * OXL(XMIT,CALFAP,CSBETA,CGAMAP) *                       RBCOM347
     1 OXL(RECV,CALFBP,CSBETB,CGAMBP)                                   RBCOM348
C                                                                       RBCOM349
      RV(KTT,KBAND,JBOUND) = RV(KTT,KBAND,JBOUND) + DELR                RBCOM350
```

SUBROUTINE RBCOMP

```
 6030 CONTINUE                                                    RBCOM351
C                                                                 RBCOM352
 7010 IF (TTMAX) GO TO 7020                                       RBCOM353
      KTIM = KTIM + K1                                            RBCOM354
      IF (KTIM .LE. JD) GO TO 4050                                RBCOM355
 7020 IF (DD(1) .LE. BAND(KBAND+1)) KBAND = KBAND + K1            RBCOM356
      IF (KBAND .GT. NBAND) GO TO 7030                            RBCOM357
      IF (JDD .NE. JDMIN) GO TO 3060                              RBCOM358
      IF (DD(1) .LE. FZRO) GO TO 7030                             RBCOM359
      DOPSKP = .TRUE.                                             RBCOM360
      DD(2) = DD(1)                                               RBCOM361
      DD(1) = AMAX1(BAND(KBAND+1), FZRO)                          RBCOM362
      GO TO 4020                                                  RBCOM363
C                                                                 RBCOM364
 7030 IF (MX .EQ. JD) GO TO 7040                                  RBCOM365
      MN = MX                                                     RBCOM366
      GO TO 3010                                                  RBCOM367
C                                                                 RBCOM368
 7040 MN = K1                                                     RBCOM369
      IF (THTMAX .LE. F90) GO TO 7060                             RBCOM370
 7050 CONTINUE                                                    RBCOM371
 7060 IF (JA .EQ. JB) GO TO 7080                                  RBCOM372
 7070 CONTINUE                                                    RBCOM373
C                                                                 RBCOM374
 7080 CONTINUE                                                    RBCOM375
 8010 CONTINUE                                                    RBCOM376
 9010 CONTINUE                                                    RBCOM377
C                                                                 RBCOM378
      REWIND BR1                                                  RBCOM379
      RETURN                                                      RBCOM380
      END                                                         RBCOM381
```

SUBROUTINE RBCOMP

SYMBOL		REFERENCES									
1010	-	38	40*								
1020	-	37	42*								
1030	-	59	61*								
1040	-	64	67*								
1050	-	75	77	80	94*						
2010	-	116	124*								
2020	-	113	126*								
3010	-	134*	367								
3020	-	138	140*								
3030	-	139	143*								
3040	-	151	153*								
3050	-	150	162*								
3060	-	164*	358								
3070	-	152	179*								
3080	-	177	191*								
3090	-	152	193*								
4010	-	189	206*								
4020	-	162	208*	363							
4030	-	213	217*								
4040	-	212	214	219*							
4050	-	224*	355								
4060	-	235*	242								
4070	-	233	239*								
4080	-	218	240	243*							
4090	-	237	245*								
4100	-	231	257*								
4110	-	230	268*								
4120	-	220	269*								
4130	-	255	266	279*							
5010	-	303	312*								
5020	-	285	313*								
6010	-	208	316*								
6020	-	316	319*								
6030	-	328	351*								
7010	-	224	317	353*							
7020	-	215	217	353	356*						
7030	-	154	357	359	365*						
7040	-	365	369*								
7050	-	107	371*								
7060	-	370	372*								
7070	-	100	373*								
7080	-	49	57	372	375*						
8010	-	33	376*								
9010	-	25	27	377*							
A	-	13C0	305=	309	311	329=	331=	347			
ABS	-	91	331								
AINTRP	-	84	85	87	196	197	198	199	200	201	202
		203	204	270	271	272	273	274	275	276	277
AMAX1	-	55	294	362							
AMIN1	-	56	229								
AMOD	-	90									
ARG	-	10C0	219=	229=	231	235	239	241	269		
ATAN2	-	333									
BAND	-	152	177	192	356	362					
BR1	-	39	42RD	43	50RD	379					
COKT	-	124	125								
CALFAP	-	12C0	342=	347							
CALFBP	-	12C0	344=	348							
CGAMAP	-	12C0	343=	347							
CGAMBP	-	12C0	345=	348							
COMTMU	-	8C0	252=	264=	276=	288					
COS	-	88	89	92	337	339					
COSA	-	12C0	321=	337	338						
COSB	-	12C0	322=	339	340						
COSOMT	-	92=	93	117	123	187	203	264	276		
COSTHA	-	88=	117	118	119	124	125	184	200	261	273
COSTHB	-	89=	185	201	262	274					
CSALFA	-	1100	337=	342	343						

SUBROUTINE RBCOMP

APPENDIX G

SUBROUTINE RBCOMP

LMKS2	-	108	109								
ML	-	17EQ	114=	117	119	125	135=	136	144	217	218
		225	226	227	239	240	243	319	320	331	
MLMU	-	16CO	17EQ	213	233						
MM	-	16CO	109=	115	236=	243=	245	246	247	248	249
		250	251	252	253						
MN	-	16CO	17EQ	127=	134	138	193	366=	369=		
MNMX	-	17EQ	19DI	144	145						
MU	-	17EQ	115=	117	118	124	136=	145	214	225	226
		227	234	235	236	241	246	247	248	258	259
		260	270	271	272	319	320	331			
MX	-	16CO	143=	193	365	366					
NA	-	16CO	42RD	56	59	62	64	76	86		
NB	-	16CO	50RD	56	79						
NBAND	-	151	357								
NBOUND	-	26									
NEXT	-	16CO	150=	162=	195	206					
NSPRN1	-	151									
OMEGA	-	90	116								
OMT	-	90=	91=	92	93	186	202	263	275		
OMTD	-	7CO	172=	186=	202=	251					
OMTTMU	-	8CO	251=	263=	275=	335					
OXL	-	347	348								
PHI	-	13CO	333=	335	337	338					
PHMOMT	-	13CO	335=	339	340						
PI	-	91									
QA	-	10CO	117=	120	121	122	288=	291	292	295	
QAABC	-	10CO	18EQ	295=	311						
QAB	-	10CO	18EQ	120=	121	291=	294	306			
QABC	-	10CO	294=	295	296						
QAC	-	10CO	292=	309							
QB	-	10CO	118=	119	120	121	122	289=	291	293	296
QBABC	-	10CO	18EQ	296=	309						
QBC	-	10CO	293=	311							
QC	-	10CO	119=	121	122	290=	292	293	294		
QCOS	-	18EQ	111=	121=	122	123	125				
QCOSM	-	18EQ	112=	123=	124						
QSIN	-	18EQ	122=	123							
R	-	85=	87=	183	199	258	270				
RBCOMP	-	1									
RBMA	-	42RD	87								
RBMB	-	50RD	85								
RBTA	-	42RD	55	56	76						
RBTB	-	50RD	55	56	79						
RBTHA	-	42RD	86								
RBTHB	-	50RD	84								
RBXA	-	42RD	60	76	86						
RBXB	-	50RD	66	79							
RD	-	6CO	169=	183=	199=	246					
RECV	-	348									
RETURN	-	380									
RT	-	9CO	225=	246=	258=	270=	331				
RV	-	350=									
SIGN	-	91	93								
SIN	-	319	320	338	340						
SINOMT	-	93=	118	123	188	204	265	277			
SNKSI	-	342	343	344	345						
SNOMTD	-	7CO	174=	188=	204=	253					
SOMTMU	-	8CO	129=	253=	265=	277=	289				
SORT	-	68									
SQRT	-	93	121	294	321	322					
T	-	76=	79=	90	180	196	229	269			
TABLKP	-	76	79	86	193	269					
TD	-	6CO	166=	180=	196=	214	217	218	235	239	241
THA	-	86=	88	181	197	259	271				
THAD	-	6CO	167=	181=	197=	247					
THAT	-	9CO	226=	247=	259=	271=	319				
THB	-	84=	89	182	198	260	272				
THBD	-	6CO	168=	182=	198=	248					
THBT	-	9CO	227=	248=	260=	272=	320				

OD 52258

SUBROUTINE RBCOMP

```
THTMAX  -   370
TIME    -    55     56
TTMAX   -    13C0   20L6   210=   241=   268=   353
TWMAX   -    13C0   56=     57    217    229    231
TWMIN   -    13C0   55=     57    214    218    219
TWOPI   -    90     91
VS      -   124    125
X       -    60=    66=    68A6    76     79     83=    86    179    193    224
            257    269
X1      -    14C0   308=   329    330    334
X2      -    14C0   329    330    334
X3      -    14C0   308    309=   329    330    334
X4      -    14C0   329    330    334
XD      -     6C0   165=   179=   193=   234    240    245
XMIT    -   347
XTMU    -     8C0   224    234    240    245=   257=   269=   306
Y1      -    14C0   310=   329    330    333
Y2      -    14C0   329    330    333
Y3      -    14C0   310    311=   329    330    333
Y4      -    14C0   329    330    333
```

APPENDIX G

```
      SUBROUTINE RVCOMP                                                 RVCOMP 1
C                                                                       RVCOMP 2
      INCLUDE DCOMN1                                                    RVCOMP 3
      INCLUDE DCOMN2                                                    RVCOMP 4
      INCLUDE DCOMN4                                                    RVCOMP 5
      COMMON I,J,K,NN,OMT,CSOMTF,SNOMTF,FR,FRSQ,FXSQ,FX,CSA,SNA,PN,B,G  RVCOMP 6
      COMMON SNPHI(2),CSPHI(2),X(2),Y(2),Z(2),D,DD,R,FRV                RVCOMP 7
      COMMON T1,T2,A1,A2,VOL1,T3                                        RVCOMP 8
C                                                                       RVCOMP 9
C         COMPUTE ONE-WAY RANGE TO MIDPOINT OF TRANSMITTED PULSE.      RVCOMP10
C         PRE-COMPUTE RELATED VALUES FOR VOLUME AND FOR RANGE LOSSES.  RVCOMP11
C                                                                       RVCOMP12
      R = TIME(KT) * CO * FPT5                                          RVCOMP13
      FRV = EXP((LOGMVI - ALPHC * TIME(KT))/F10 * FLOG10)/R**4          RVCOMP14
      IF (OMEGA .EQ. O.) FRV = FRV *                                    RVCOMP15
     1     ((R + DR)**3 - AMAX1(R - DR, FO)**3) * P123                  RVCOMP16
      IF (VPTTRN) GO TO 9C00                                            RVCOMP17
C                                                                       RVCOMP18
C         PATTERN LOSSES MUST BE COMPUTED.                              RVCOMP19
C                                                                       RVCOMP20
      OMT = AMOD(OMEGA * TIME(KT), TWOPI)                               RVCOMP21
      IF (ABS(OMT) .GT. PI) OMT = OMT - SIGN(TWOPI, OMT)                RVCOMP22
      CSOMTF = COS(OMT)                                                 RVCOMP23
      SNOMTF = SIGN(SQRT(F1-CSOMTF**2),OMT) * FZRO                      RVCOMP24
      CSOMTF = CSOMTF * FZ2                                             RVCOMP25
      FCSGAM(NSPRH1) = O.                                               RVCOMP26
      IF (OMEGA .EQ. O.) FCSGAM(NSPRH1) = F1                            RVCOMP27
      T1 = AMAX1(TIME(KT) - DELT, FO) * FPT5                            RVCOMP28
      T2 = ((TIME(KT) + DELT) * FPT5 - T1)/F10                          RVCOMP29
C                                                                       RVCOMP30
C         COMPUTE AVERAGE PATTERN LOSSES FOR EACH BAND.                 RVCOMP31
C                                                                       RVCOMP32
      DO 8000 I = NSPRH1, NBAND                                         RVCOMP33
      FGAM(I) = O.                                                      RVCOMP34
      FR = (BAND(I) + BAND(I+1)) * FPT5                                 RVCOMP35
      FRSQ = FR**2                                                      RVCOMP36
      FXSQ = FZSQ + FRSQ + FR * CSOMTF                                  RVCOMP37
      FX = SQRT(FXSQ)                                                   RVCOMP38
      A1 = FCOVS * (FR - FZRO)                                          RVCOMP39
      CSA = A1/FX                                                       RVCOMP40
      IF (ABS(CSA) .GT. F1) CSA = SIGN(F1, CSA)                         RVCOMP41
      SNA = SQRT(F1 - CSA**2)                                           RVCOMP42
      NN = MINO(K2 * IFIX(FNBAND * SNA) + K1, 360)                      RVCOMP43
      PN = NN                                                           RVCOMP44
      DD = TWOPI/PN                                                     RVCOMP45
      D = O.                                                            RVCOMP46
      SNPHI(1) = SNOMTF/FX                                              RVCOMP47
      SNPHI(2) = -SNPHI(1) * FR/FZRO                                    RVCOMP48
      CSPHI(1) = SQRT(F1 - SNPHI(1)**2)                                 RVCOMP49
      CSPHI(2) = SQRT(F1 - SNPHI(2)**2)                                 RVCOMP50
      IF (ABS(FZSQ - FRSQ) .LE. FXSQ) GO TO 2000                        RVCOMP51
      CSPHI(1) = SIGN(CSPHI(1), CSA)                                    RVCOMP52
      CSPHI(2) = -SIGN(CSPHI(1), CSA)                                   RVCOMP53
 2000 DO 4000 J = 1,NN                                                  RVCOMP54
      D = D + DD                                                        RVCOMP55
      B = SNA * SIN(D)                                                  RVCOMP56
      G = SNA * COS(D)                                                  RVCOMP57
C                                                                       RVCOMP58
      DO 3000 K = 1, 2                                                  RVCOMP59
      X(K) = CSA * CSPHI(K) - B * SNPHI(K)                              RVCOMP60
      Y(K) = B * CSPHI(K) + CSA * SNPHI(K)                              RVCOMP61
      X(K) = X(K) * CSKSI + G * SNKSI                                   RVCOMP62
      Z(K) = G * CSKSI - X(K) * SNKSI                                   RVCOMP63
 3000 CONTINUE                                                          RVCOMP64
      FGAM(I) = FGAM(I) + OXL(XMIT,X(1),Y(1),Z(1))                      RVCOMP65
     1                  * OXL(RECV,X(2),Y(2),Z(2))                      RVCOMP66
 4000 CONTINUE                                                          RVCOMP67
      FGAM(I) = FGAM(I)/PN * EXPS                                       RVCOMP68
      IF (OMEGA .EQ. O.) GO TO 6000                                     RVCOMP69
      T3 = T1                                                           RVCOMP70
```

OD 52258

SUBROUTINE RVCOMP

```
      A2 = FZ2 * FR                                                     RVCOMP71
      A1 = A1/SQRT(A2)                                                  RVCOMP72
      A2 = (FRSQ + FZSQ)/A2                                             RVCOMP73
      VOL1 = 0.                                                         RVCOMP74
C                                                                       RVCOMP75
      DO 5000 J = 1, 10                                                 RVCOMP76
      T3 = T3 + T2                                                      RVCOMP77
      VOL1 = VOL1 - T3**2 * AMAX1(F1 - A1/SQRT(A2 + COS(OMEGA * T3)),F0)RVCOMP78
 5000 CONTINUE                                                          RVCOMP79
      FCSGAM(I+1) = VOL1 * T2 * FC03                                    RVCOMP80
      GOTO 7000                                                         RVCOMP81
 6000 FCSGAM(I+1) =  FCOVS * (BAND(I+1) - FZRO)/(BAND(I+1) + FZRO)      RVCOMP82
 7000 FGAM(I) = FGAM(I) * (FCSGAM(I) - FCSGAM(I+1))                     RVCOMP83
      RVV(KTT,I) = FRV * FGAM(I)                                        RVCOMP84
 8000 CONTINUE                                                          RVCOMP85
C                                                                       RVCOMP86
C          FOR STRAIGHT-RUNNING CASE, PATTERN LOSSES FOR ANY BAND WILL  RVCOMP87
C          BE THE SAME AT ALL TIMES.                                    RVCOMP88
C                                                                       RVCOMP89
      IF (OMEGA .EQ. 0.) VPTTRN = .TRUE.                                RVCOMP90
      GO TO 30000                                                       RVCOMP91
 9000 DO 10000 I = NSPRH1, NBAND                                        RVCOMP92
      RVV(KTT,I) = FRV * FGAM(I)                                        RVCOMP93
10000 CONTINUE                                                          RVCOMP94
30000 RETURN                                                            RVCOMP95
      END                                                               RVCOMP96
```

APPENDIX G

SUBROUTINE RVCOMP

SYMBOL		REFERENCES									
2000	-	51	54*								
3000	-	59	64*								
4000	-	54	67*								
5000	-	76	79*								
6000	-	69	82*								
7000	-	81	83*								
8000	-	33	85*								
9000	-	17	92*								
10000	-	92	94*								
30000	-	91	95*								
A1	-	8C0	39=	40	72=	78					
A2	-	8C0	71=	72	73=	78					
ABS	-	22	41	51							
ALPHC	-	14									
AMAX1	-	16	28	78							
AMOD	-	21									
B	-	6C0	56=	60	61						
BAND	-	35	82								
C0	-	13									
COS	-	23	57	78							
CSA	-	6C0	40=	41=	42	52	53	60	61		
CSKSI	-	62	63								
CSOMTF	-	6C0	23=	24	25=	37					
CSPHI	-	7C0	49=	50=	52=	53=	60	61			
D	-	7C0	46=	55=	56	57					
DCOMN1	-	3									
DCOMN2	-	4									
DCOMN4	-	5									
DD	-	7C0	45=	55							
DELT	-	28	29								
DR	-	16									
EXP	-	14									
EXPS	-	68									
F0	-	16	28	78							
F1	-	24	27	41	42	49	50	78			
F1C	-	14	29								
FC03	-	80									
FCOVS	-	39	82								
FCSGAM	-	26=	27=	80=	82=	83					
FGAM	-	34=	65=	68=	83=	84	93				
FLOG10	-	14									
FNBAND	-	43									
FPTS	-	13	28	29	35						
FR	-	6C0	35=	36	37	39	48	71			
FRSQ	-	6C0	36=	37	51	73					
FRV	-	7C0	14=	15=	84	93					
FX	-	6C0	38=	40	47						
FXSQ	-	6C0	37=	38	51						
FZ2	-	25	71								
FZRO	-	24	39	48	82						
FZSQ	-	37	51	73							
G	-	6C0	57=	62	63						
I	-	6C0	33=	34	35	65	68	80	82	83	84
		92=	93								
IFIX	-	43									
J	-	6C0	54=	76=							
K	-	6C0	59=	60	61	62	63				
K1	-	43									
K2	-	43									
KT	-	13	14	21	28	29					
KTT	-	84	93								
LOGMVI	-	14									
MINO	-	43									
NBAND	-	33	92								
NN	-	6C0	43=	44	54						
NSPRN1	-	26	27	33	92						
OMEGA	-	15	21	27	69	78	90				
OMT	-	6C0	21=	22=	23	24					

OD 52258

APPENDIX G

```
      SUBROUTINE RVSPRD                                       RVSPRD 1
C                                                             RVSPRD 2
      INCLUDE DCOMN1                                          RVSPRD 3
      INCLUDE DCOMN2                                          RVSPRD 4
      COMMON IMAX,I,II,J,K,KK                                 RVSPRD 5
C                                                             RVSPRD 6
      DO 13 J = K1, K3                                        RVSPRD 7
      DO 11 I = NSPRN1, NBAND                                 RVSPRD 8
      II = I-NSPRN                                            RVSPRD 9
      DO 10 K = K1, NSPR1                                     RVSPRD10
      KK = ABS(K-NSPRN1) + K1                                 RVSPRD11
      RV(2,II,J) = RV(2,II,J) + RV(1,I,J) * SPRED(KK,J)       RVSPRD12
      II = II + 1                                             RVSPRD13
       IF (II .GT. LNBAND) GO TO 11                           RVSPRD14
   10 CONTINUE                                                RVSPRD15
   11 CONTINUE                                                RVSPRD16
   13 CONTINUE                                                RVSPRD17
      RETURN                                                  RVSPRD18
      END                                                     RVSPRD19
```

OD 52258

SUBROUTINE RVSPRD

SYMBOL		= = = = = = = = = = = = = = = =	REFERENCES	= = = = = = = = = = = = = = = =	
10	-	10	15*		
11	-	8	14	16*	
13	-	7	17*		
ABS	-	11			
BCOMN1	-	3			
BCOMN2	-	4			
I	-	5CO	8=	9	12
II	-	5CO	9=	12	13= 14
IMAX	-	5CO			
J	-	5CO	7=	12	
K	-	5CO	10=	11	
K1	-	7	10	11	
K3	-	7			
KK	-	5CO	11=	12	
LMBAND	-	14			
NBAND	-	8			
NSPR1	-	10			
NSPRH	-	9			
NSPRH1	-	8	11		
RETURN	-	18			
RV	-	12=			
RVSPRD	-	1			
SPRED	-	12			

-+-

APPENDIX G

```
      SUBROUTINE RTCOMP
C
      INCLUDE DCOMN1
      INCLUDE DCOMN2
      COMMON IMAX,I,J,T,TR
C
      IMAX = KTT
      IF (.NOT. SPREAD) GO TO 2
      IMAX = K2
      IF (.NOT. (FILTER .OR. TVG)) GO TO 2
      IMAX = K3
      T = F1
      IF (TVG) T = TVGF(TIME(KT))
      DO 1 I = 1, MNBND
      TR = T
      IF (FILTER) TR = T * RRFS(I)
      RVS(3,I) = RVS(2,I) * TR
      RVB(3,I) = RVB(2,I) * TR
      RVV(3,I) = RVV(2,I) * TR
    1 CONTINUE
C
    2 DO 5 I = KTT, IMAX
      J = 0
C
    3 J = J + K1
      RVS(I,LMBND1) = RVS(I,LMBND1) + RVS(I,J)
      RVB(I,LMBND1) = RVB(I,LMBND1) + RVB(I,J)
      RVV(I,LMBND1) = RVV(I,LMBND1) + RVV(I,J)
C
    4 RVT(I,J) = RVS(I,J) + RVB(I,J) + RVV(I,J)
      IF (RVS(I,J) .NE. 0.) RVS(I,J) = F10 * ALOG10(RVS(I,J))
      IF (RVB(I,J) .NE. 0.) RVB(I,J) = F10 * ALOG10(RVB(I,J))
      IF (RVV(I,J) .NE. 0.) RVV(I,J) = F10 * ALOG10(RVV(I,J))
      IF (RVT(I,J) .NE. 0.) RVT(I,J) = F10 * ALOG10(RVT(I,J))
C
      IF (J .LT. MNBND) GO TO 3
      IF (J .EQ. LMBND1) GO TO 5
      J = LMBND1
      GO TO 4
C
    5 CONTINUE
      RETURN
      END
```

SUBROUTINE RTCOMP

SYMBOL		REFERENCES									
1	-	14	20*								
2	-	8	10	22*							
3	-	25*	36								
4	-	30*	39								
5	-	22	37	41*							
ALOG10	-	31	32	33	34						
DCOMN1	-	3									
DCOMN2	-	4									
F1	-	12									
F10	-	31	32	33	34						
FILTER	-	10	16								
I	-	5CO	14=	16	17	18	19	22=	26	27	28
		30	31	32	33	34					
IMAX	-	5CO	7=	9=	11=	22					
J	-	5CO	23=	25=	26	27	28	30	31	32	33
		34	36	37	38=						
K1	-	25									
K2	-	9									
K3	-	11									
KT	-	13									
KTT	-	7	22								
LMBND1	-	26	27	28	37	38					
MNBND	-	14	36								
RETURN	-	42									
RRFS	-	16									
RTCOMP	-	1									
RVB	-	16=	27=	30	32=						
RVS	-	17=	26=	30	31=						
RVT	-	30=	34=								
RVV	-	19=	28=	30	33=						
SPREAD	-	8									
T	-	5CO	12=	13=	15	16					
TIME	-	13									
TR	-	5CO	15=	16=	17	18	19				
TV6	-	10	13								
TV6F	-	13									

APPENDIX G

```
      SUBROUTINE RVPRNT
C
      INCLUDE DCOMN1
      INCLUDE DCOMN2
      COMMON I,J,K,L,II
      DIMENSION HRVB(6)
      DATA HTIME/6HTIME =/,(HRVB(I), I = 1,6)
     1 /6H  SUR, 6HFACE  , 6HBOTTOM, 6H  VOL, 6HUME  , 6H TOTAL/
      DATA ITOTAL/1/
C
      IF (PLOT) WRITE (IPLT) TIME(NTMIN),MNBND,(BNDOUT(K),BNDOUT(K+1),
     1              RVS(2,K),RVB(2,K),RVV(2,K),RVT(2,K), K = 1, MNBND)
      IF (NOPRNT) GO TO 30000
      IF (TOTALS) GO TO 2000
      J = NPSTRT
    1 PAGE = PAGE + K1
      WRITE (IPRT,10) HED,IDC,IDV,IDATE,PAGE,NPAGE,
     1                VS,CO,FZRO,DO,S,KSID,OMEGAD,PING,DELT2,DBTTM
      IF (TOTALS) GO TO 2020
C
      IF (SPREAD) GO TO 5
      WRITE (IPRT,11) HBAND,(HTIME,TIME(K), K = NTMIN, NTMAX)
      GO TO 6
C
    5 WRITE (IPRT,11) HBAND,(HTIME,TIME(NTMIN), K = 1, I)
C
    6 WRITE (IPRT,12) HOUTPT, (HRVB, K = 1, I)
      IF (TOTALS) GO TO 2100
      DO 3 K = K1, K8
      WRITE (IPRT,20)
C
      DO 2 L = K1, K5
      WRITE (IPRT, 20) BNDOUT(J), BNDOUT(J+1),
     1     (RVS(II,J), RVB(II,J), RVV(II,J), RVT(II,J), II = 1, I)
      J = J + IPEVRY
      IF (J .GT. MNBND) GO TO 4
C
    2 CONTINUE
    3 CONTINUE
      GO TO 1
C
    4 WRITE (IPRT,30) HTOT1, (RVS(II,LMBND1),RVB(II,LMBND1),
     1                 RVV(II,LMBND1),RVT(II,LMBND1), II = 1, I)
      GO TO 30000
C
 2000 IF (.NOT. SPREAD) I = K1
      J = K1
      II = NTMIN
 2010 ITOTAL = ITOTAL - K1
      IF (ITOTAL .NE. 0) GO TO 2100
      ITOTAL = K40
      GO TO 1
C
 2020 WRITE (IPRT,13) HTOT1,MNBND,HTOT2,BWIDTH,HTOT3
      GO TO 6
C
 2100 IF (MOD(ITOTAL,K5) .EQ. 0) WRITE (IPRT,20)
      WRITE (IPRT,21) TIME(II), (RVS(K,LMBND1),RVB(K,LMBND1),
     1                       RVV(K,LMBND1),RVT(K,LMBND1), K = J, I)
      I = I + K1
      J = J + K1
      II = II + K1
      IF (II .LE. NTMAX) GO TO 2010
C
30000 RETURN
C
   10 FORMAT (8A6,A4,9H   IDC   ,A6,6X,6HIDV   , A6,7X,4HDATE,2A6,7X,
     1 4HPAGE,I5,3H OF,I5/
     2132HCV.S, KTS.   C.O, YDS./SEC.   F.O, KHZ.   D.O, FT.   S, DB.
     3XI, DEG.   OMEGA, DEG./SEC.   P.I., SEC.   DEL. T, SEC.   D. BTM.,
```

APPENDIX G

SUBROUTINE RVPRNT

```
     4 FT./ F8.2,F15.2,F15.4,F13.4,F9.2,F10.3,F15.3,F16.4,F13.4,F17.4)  RVPRNT71
C                                                                        RVPRNT72
   11 FORMAT (1H0, 2A6, A4, 3(13X,A6,F12.8,7X))                          RVPRNT73
C                                                                        RVPRNT74
   12 FORMAT (5X,2A6,17X,2A6,23X,3A6,13X,5A6/3X,2A6,3X,3(2X,6A6))         RVPRNT75
C                                                                        RVPRNT76
   13 FORMAT (1H0,30X,4A6,14,2A6,A3,F6.3,3A6)                            RVPRNT77
C                                                                        RVPRNT78
   20 FORMAT (1X,F8.4,F9.4,3(2X,4F9.2))                                  RVPRNT79
C                                                                        RVPRNT80
   21 FORMAT (F17.8,1X,3(2X,4F9.2))                                      RVPRNT81
C                                                                        RVPRNT82
   30 FORMAT (1H0,3A6,A1,4F9.2,2(2X,4F9.2))                              RVPRNT83
      END                                                                RVPRNT84
```

APPENDIX G

SUBROUTINE RVPRNT

SYMBOL		= = = = = = = = = = = = = = = = REFERENCES = = = = = = = = = = = = = = = =									
1	-	16*	40	52							
2	-	32	38*								
3	-	29	39*								
4	-	36	42*								
5	-	21	25*								
6	-	23	27*	55							
10	-	17WR	67*								
11	-	22WR	25WR	73*							
12	-	27WR	75*								
13	-	54WR	77*								
20	-	30WR	33WR	57WR	79*						
21	-	58WR	81*								
30	-	42WR	83*								
2000	-	14	46*								
2010	-	49*	63								
2020	-	19	54*								
2100	-	28	50	57*							
30000	-	13	44	65*							
BNDOUT	-	11WR	33WR								
BWIDTH	-	54WR									
CO	-	18WR									
DO	-	18WR									
DBTTM	-	18WR									
DCOMN1	-	3									
DCOMN2	-	4									
DELT2	-	18WR									
FZRO	-	18WR									
HBAND	-	22WR	25WR								
HED	-	17WR									
HOUTPT	-	27WR									
HRVD	-	6DI	7DA	27WR							
HTIME	-	7DA	22WR	25WR							
HTOT1	-	42WR	54WR								
HTOT2	-	54WR									
HTOT3	-	54WR									
I	-	5CO	7DA	25WR	27WR	34WR	43WR	46=	59WR	60=	
IDATE	-	17WR									
IDC	-	17WR									
IDV	-	17WR									
II	-	5CO	34WR	42WR	43WR	48=	58WR	62=	63		
IPEVRY	-	35									
IPLT	-	11WR									
IPRT	-	17WR	22WR	25WR	27WR	30WR	33WR	42WR	54WR	57WR	58WR
ITOTAL	-	9DA	49=	50	51=	57					
J	-	5CO	15=	33WR	34WR	35=	36	47=	59WR	61=	
K	-	5CO	11WR	12WR	22WR	25WR	27WR	29=	58WR	59WR	
K1	-	16	29	32	46	47	49	60	61	62	
K40	-	51									
K5	-	32	57								
K8	-	29									
KSID	-	18WR									
L	-	5CO	32=								
LMBND1	-	42WR	43WR	58WR	59WR						
MNBND	-	11WR	12WR	36	54WR						
MOD	-	57									
NOPRNT	-	13									
NPAGE	-	17WR									
NPSTRT	-	15									
NTMAX	-	22WR	63								
NTMIN	-	11WR	22WR	25WR	48						
OMEGAD	-	18WR									
PAGE	-	16=	17WR								
PING	-	18WR									
PLOT	-	11									
RETURN	-	65									
RVD	-	12WR	34WR	42WR	58WR						
RVPRNT	-	1									
RVS	-	12WR	34WR	42WR	58WR						

APPENDIX G

SUBROUTINE RVPRNT

RVT	-	12WR	34WR	43WR	59WR
RVV	-	12WR	34WR	43WR	59WR
S	-	18WR			
SPREAD	-	21	46		
TIME	-	11WR	22WR	25WR	58WR
TOTALS	-	14	19	28	
VS	-	18WR			

BLOCK DATA FOR DOP

```
      BLOCK DATA                                                        BLOCK  1
C                                                                       BLOCK  2
      INCLUDE DCOMN1                                                    BLOCK  3
      INCLUDE DCOMN6                                                    BLOCK  4
C                                                                       BLOCK  5
      DATA  K0/0/,K1/1/,K2/2/,K3/3/,K5/5/,K6/6/,K8/8/,K10/10/,K40/40/   BLOCK  6
      DATA  (TRS(I), I = 1,6)/.01,.05,.1,.2,.3,.4/,FPT5/.5/,            BLOCK  7
     1  (TRS2(I), I = 1,4)/.6,.7,.8,.9/,F1/1./,                         BLOCK  8
     2  (TRS3(I), I = 1,4)/1.2,1.4,1.6,1.8/,F2/2./,F4/4./,F10/10./,     BLOCK  9
     3  F20/20./,FLOG10/2.30258509/,INFNT/1.E38/                        BLOCK 10
     4 PI23/2.09439510/,PI/3.14159265/,TWOPI/6.28318531/,               BLOCK 11
     5  DEGRAD/.0174532925/,SHIFT/01504000000000/,VDKT/1.77625736/,     BLOCK 12
     6 F1E3/1000./,LOG4PI/10.9920986/,F1MIN/02007777777777/,F3/3./,     BLOCK 13
     7 F90/90./,F180/180./                                              BLOCK 14
      DATA AR1/22/,BR1/28/,INPT/5/,IPRT/6/,IPLT/15/                     BLOCK 15
      DATA HED/52H1PROGRAM 800003 -- DOPPLER CONTENT OF REVERBERATION / BLOCK 16
      DATA (HEADS(I), I = 1, 41)/30H  DOPPLER BAND,     KNOTS     ,     BLOCK 17
     1                           30H FREQUENCY BAND,  KILOHERTZ   ,     BLOCK 18
     2                           30H PURE TONE   AFTER SPREADING  ,     BLOCK 19
     3                           30H                              ,     BLOCK 20
     4                           30H       SPREAD WITH FILTER     ,     BLOCK 21
     5                           30H          SPREAD WITH TVG     ,     BLOCK 22
     6                           30H  SPREAD WITH FILTER AND TVG  ,     BLOCK 23
     7                        36H KNOTS HERTZFROM-     TO-     TIME /   BLOCK 24
      DATA (HTOT1(I), I = 1, 4)/24HTOTAL REVERBERATION FROM/            BLOCK 25
      DATA (HTOT2(I), I = 1, 3)/18H BANDS, EACH OF   /                  BLOCK 26
      DATA (HTOT3(I), I = 1, 3)/18H       BANDWIDTH  /                  BLOCK 27
      DATA ITABMN/1/,IPEVRY/1/                                          BLOCK 28
C                                                                       BLOCK 29
      DATA             NNAMES/40/,NAMCNT/24 * 0/,LFLAGS/16 * .FALSE./   BLOCK 30
      DATA             ((NAMDAT(I,J), J = 1, 3), I = 1, 40)/            BLOCK 31
     A                 6HIDC   ,  0, 4,6HDATE  ,  1, 2,6HIDV   , 3, 1,BLOCK 32
     B                 6HD0    ,  4, 1,6HC0    ,  5, 1,6HALPHC , 6, 1,BLOCK 33
     C                 6HPING  ,  7, 1,6HDBTTM ,  8, 1,6HLOGMV , 9, 1,BLOCK 34
     D                 6HS     , 10, 1,6HXI    , 11, 1,6HVS    ,12, 1,BLOCK 35
     E                 6HBWIDTH, 13, 1,6HDELT  , 14, 1,6HF0    ,15, 1,BLOCK 36
     F                 6HNBEAM , 16, 1,6HOMEGA , 17, 1,6HTHTMAX, 18, 1,BLOCK 37
     G                 6HPULSE , 19, 1,6HPRINTE, 20, 1,                 BLOCK 38
     H                 6HTIME  ,21,400,6HSSPRED,421,150,6HBSPRED,571,150,BLOCK 39
     I                 6HVSPRED,721,150,                                BLOCK 40
     J                 6HCENTER,  0, 1,6HGO    ,  0, 1,6HEND   ,  0,1,BLOCK 41
     K                 6HFILTER,  0, 1,6HKNOTS ,  0, 1,6HNOBOTT,  0,1,BLOCK 42
     L                 6HNOPRIN,  0, 1,6HNOSURF,  0, 1,6HNOTAPE,  0,1,BLOCK 43
     M                 6HNOVOLU,  0, 1,6HPLOT  ,  0, 1,6HSPREAD,  0,1,BLOCK 44
     N                 6HTIMECO,  0, 1,6HTOTALS,  0, 1,6HTVG   ,  0,1,BLOCK 45
     O                 6HRELATI,  0, 1/                                 BLOCK 46
      END                                                               BLOCK 47
```

OD 52258

BLOCK DATA FOR DOP

```
SYMBOL    - - - - - - - - - - - - - - - -   REFERENCES   - - - - - - - - - - - - - - - -
AR1      -    15DA
BR1      -    15DA
BCOMN1   -    3
BCOMN6   -    4
DEGRAD   -    12DA
F1       -    8DA
F10      -    9DA
F180     -    14DA
F1E3     -    13DA
F1MIN    -    13DA
F2       -    9DA
F20      -    10DA
F3       -    13DA
F4       -    9DA
F90      -    14DA
FLOG10   -    10DA
FPTS     -    7DA
HEADS    -    17DA
HED      -    16DA
HTOT1    -    25DA
HTOT2    -    26DA
HTOT3    -    27DA
I        -    7DA     8DA     9DA     17DA     25DA     26DA     27DA     31DA
INFNT    -    10DA
INPT     -    15DA
IPEVRY   -    28DA
IPLT     -    15DA
IPRT     -    15DA
ITABNN   -    28DA
J        -    31DA
K0       -    6DA
K1       -    6DA
K10      -    6DA
K2       -    6DA
K3       -    6DA
K40      -    6DA
K5       -    6DA
K6       -    6DA
K8       -    6DA
LFLAGS   -    30DA
LOG4PI   -    13DA
NAMCNT   -    30DA
NAMDAT   -    31DA
NNAMES   -    30DA
PI       -    11DA
PI23     -    11DA
SHIFT    -    12DA
TRS      -    7DA
TRS2     -    8DA
TRS3     -    9DA
TWOPI    -    11DA
YDKT     -    12DA
1BLCK    -    1
```

APPENDIX G

```
      FUNCTION FNCNS(ARG)
C
      ENTRY OXL (IFLAG,COSA,COSB,COSC)
C
C         NBEAM = 0 FOR NARROW BEAM
C               = 1 FOR BROAD BEAM
C         IFLAG = 0 FOR RECEIVE
C               = 1 FOR TRANSMIT
C
      COMMON /CINPUT/ X(16), NBEAM
C
      A = ACOS(COSA)
      FCNS = 10**(-2.*A)
      IF (IFLAG .EQ. 0) GO TO 1
      FCNS = FCNS * COS(A*2.0)**2
      IF (NBEAM .NE. 0) RETURN
      FCNS = FCNS * ABS(COSA)
    1 FCNS = FCNS * COS(A*4.0)**2
      RETURN
C
      ENTRY RRF(FREQ)
C
      FNCNS = 1. - (FREQ/31.4159)**4
      RETURN
C
      ENTRY TVGF(TIME)
C
      FNCNS = 0.1 + .9 * TIME/2.
      RETURN
      END
```

OD 52258

FUNCTION FNCNS(ARG)

SYMBOL		* * * * * * * * REFERENCES * * * * * * * *			
1	-	14	18*		
A	-	12=	13	15	18
ABS	-	17			
ACOS	-	12			
ARG	-	1AG			
CINPUT	-	10CL			
COS	-	15	18		
COSA	-	3	12	17	
COSB	-	3			
COSC	-	3			
FCNS	-	13=	15=	17=	18=
FNCNS	-	1	23=	28=	
FREQ	-	21	23		
IFLAG	-	3	14		
NBEAM	-	10CO	16		
OXL	-	3			
RETURN	-	16	19	24	29
RRF	-	21			
TIME	-	26	28		
TVGF	-	26			
X	-	10CO			

INDEX PAGE 47

```
      SUBROUTINE SORT(ARRAY,LENGTH)                                     SORT   1
      DIMENSION ARRAY(1)                                                SORT   2
C                                                                       SORT   3
C           SORT IN ASCENDING ORDER (SIMPLE REPLACEMENT SORT)           SORT   4
C                                                                       SORT   5
      IF (LENGTH .LE.1) GO TO 4                                         SORT   6
      DO 2 J = 2,LENGTH                                                 SORT   7
      IF (ARRAY(J) .GE. ARRAY(J-1)) GO TO 2                             SORT   8
      TEMP = ARRAY(J)                                                   SORT   9
      I = J                                                             SORT  10
C                                                                       SORT  11
    1 ARRAY(I) = ARRAY(I-1)                                             SORT  12
      I = I - 1                                                         SORT  13
      IF ((I .GT. 1) .AND. (TEMP .LT. ARRAY(I-1))) GO TO 1              SORT  14
      ARRAY(I) = TEMP                                                   SORT  15
    2 CONTINUE                                                          SORT  16
C                                                                       SORT  17
C           REMOVE DUPLICATE ENTRIES                                    SORT  18
C                                                                       SORT  19
      DO 3 I = 2, LENGTH                                                SORT  20
      IF (ARRAY(I) .EQ. ARRAY(I-1)) GO TO 5                             SORT  21
    3 CONTINUE                                                          SORT  22
    4 RETURN                                                            SORT  23
C                                                                       SORT  24
    5 J = LENGTH                                                        SORT  25
      LENGTH = I - 1                                                    SORT  26
    6 IF (I .GE. J) GO TO 4                                             SORT  27
      I = I + 1                                                         SORT  28
      IF (ARRAY(I) .EQ. ARRAY(I-1)) GO TO 6                             SORT  29
      LENGTH = LENGTH + 1                                               SORT  30
      ARRAY(LENGTH) = ARRAY(I)                                          SORT  31
      GO TO 6                                                           SORT  32
C                                                                       SORT  33
      END                                                               SORT  34
```

OD 52258

```
I N D E X                                                                        PAGE  48
                                    SUBROUTINE SORT(ARRAY,LENGTH)

SYMBOL      * * * * * * * * * * * * * * *   REFERENCES  * * * * * * * * * * * * * * *

1        -    12*     14
2        -     7       8      16*
3        -    20      22*
4        -     6      23*     27
5        -    21      25*
6        -    27*     29      32
ARRAY    -    1AG     2DI      8       9      12=     14      15=     21      29      31=
I        -    10=     12      13=     14      15      20=     21      26      27      28=
              29      31
J        -     7=      8       9      10      25=     27
LENGTH   -    1AG      6       7      20      25      26=     30=     31
RETURN   -    23
SORT     -     1
TEMP     -     9=     14      15
```

APPENDIX G

```
      FUNCTION TABLKP(ARG,INDEP,DEPND,ITABMN,ITABMX)                    TABLKP 1
C                                                                       TABLKP 2
      COMMON /CTBLKP/ DELIND,DELDEP,FACTOR,ITABLE                       TABLKP 3
      DIMENSION INDEP(1),DEPND(1)                                       TABLKP 4
      REAL INDEP                                                        TABLKP 5
C                                                                       TABLKP 6
      DELIND = INDEP(2) - INDEP (1)                                     TABLKP 7
      DO 3 ITABLE = ITABMN, ITABMX                                      TABLKP 8
      FACTOR = ARG - INDEP(ITABLE)                                      TABLKP 9
      IF (FACTOR * DELIND) 5, 6, 3                                      TABLKP10
    3 CONTINUE                                                          TABLKP11
    4 ITABLE = 0                                                        TABLKP12
      GO TO 7                                                           TABLKP13
C                                                                       TABLKP14
    5 IF (ITABLE .EQ. 1) GO TO 4                                        TABLKP15
      DELIND = INDEP(ITABLE) - INDEP(ITABLE-1)                          TABLKP16
      FACTOR = FACTOR/DELIND                                            TABLKP17
C                                                                       TABLKP18
    6 TABLKP = AINTRP(DEPND)                                            TABLKP19
    7 RETURN                                                            TABLKP20
      END                                                               TABLKP21
```

```
I N D E X                                                                         PAGE  50
                              FUNCTION TABLKP(ARG,INDEP,DEPND,ITABMN,ITABMX)

SYMBOL          = = = = = = = = = = = = = = =    REFERENCES    = = = = = = = = = = = = = = =

3       -       8       10      11*
4       -       12*     15
5       -       10      15*
6       -       10      19*
7       -       13      20*
AINTRP  -       19
ARG     -        1AG     9
CTBLKP  -        3CL
DELDEP  -        3CO
DELIND  -        3CO     7=     10      16=     17
DEPND   -        1AG    4DI     19
FACTOR  -        3CO     9=     10      17=
INDEP   -        1AG    4DI     5RL      7       9      16
ITABLE  -        3CO     8=      9      12=     15      16
ITABMN  -        1AG     8
ITABMX  -        1AG     8
RETURN  -       20
TABLKP  -        1      19=

-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
      FUNCTION AINTRP(DEPND)                                    AINTRP 1
C                                                               AINTRP 2
      COMMON /CTBLKP/ DELIND,DELDEP,FACTOR,ITABLE               AINTRP 3
      DIMENSION DEPND(1)                                        AINTRP 4
C                                                               AINTRP 5
      IF (FACTOR .NE. 0.) GO TO 1                               AINTRP 6
      AINTRP = DEPND(ITABLE)                                    AINTRP 7
      GO TO 2                                                   AINTRP 8
C                                                               AINTRP 9
    1 DELDEP = DEPND(ITABLE) - DEPND(ITABLE-1)                  AINTRP10
      AINTRP = DEPND(ITABLE) + DELDEP * FACTOR                  AINTRP11
    2 RETURN                                                    AINTRP12
      END                                                       AINTRP13
```

I N D E X   PAGE 52

FUNCTION AINTRP(DEPND)

SYMBOL		REFERENCES				
1	-	6	10=			
2	-	8	12=			
AINTRP	-	1	7=	11=		
CTBLKP	-	3CL				
DELDEP	-	3CO	10=	11		
DELIND	-	3CO				
DEPND	-	1AG	4DI	7	10	11
FACTOR	-	3CO	6	11		
ITABLE	-	3CO	7	10	11	
RETURN	-	12				

```
      SUBROUTINE INPUT(NAMTAB,DATA,INPFLG,OUTFLG)
C
      COMMON /CINCOD/ INAM,IIMAGE,IMAGE(80),IBLANK,NUMBER(21)
      EQUIVALENCE (IQUOTE,NUMBER(17)),(ICOMMA,NUMBER(20))
      EQUIVALENCE (IEQUAL,NUMBER(21))
      COMMON /CMACHN/ NWORD,NCHAR,MAXCOL
C
      LOGICAL BCDFLG,DECPT,EXPNT,CXBOTH,OUTFLG,LVALUE,LTRUE
      DOUBLE PRECISION  VALUE,D10
      DIMENSION NAMTAB(1),DATA(1),AVALUE(1)
      EQUIVALENCE (VALUE,AVALUE,IVALUE,LVALUE,NAME),(I,IIMAGE),
     1    (NTRUE,LTRUE)
      DATA ITYPE/0/,J/3/,ITRUE/5HTRUE./,D10/1.D1/,FRMAT1/6H(80A1)/,
     1    LTRUE/.TRUE./
C
C          ITYPE = 1, INTEGER (NO DECIMAL POINT)
C          ITYPE = 2, REAL (WITH OR WITHOUT EXPONENT)
C          ITYPE = 3, DOUBLE PRECISION (D-TYPE EXPONENT)
C          ITYPE = 4, COMPLEX (TWO REAL NUMBERS IN PARENTHESES)
C          ITYPE = 5, LOGICAL (.TRUE. OR .FALSE. ONLY)
C          ITYPE = 6, ALPHAMERIC (ENCLOSED IN APOSTROPHES, THUS 'ABC')
C
      GO TO (1260, 1600, 1030, 2000), J
 1030 READ FRMAT1, IMAGE
      IF (OUTFLG) PRINT 2001, IMAGE
      I = 0
      GO TO 1600
 1100 I = I + 1
      IF (ITYPE .EQ. 6) GO TO 1160
      IF (IMAGE(I) .EQ. IBLANK) GO TO 1900
      IF (IMAGE(I) .NE. ICOMMA) GO TO (1130, 1110), J
 1105 GO TO (1200, 1400), J
 1110 DO 1120 K = 1, 19
      IF (IMAGE(I) .NE. NUMBER(K)) GO TO 1120
      IF (K .LT. 18) GO TO 1300
      IF (DECPT) GO TO 1300
 1120 CONTINUE
      IF (L + NAME .NE. 0) GO TO 1995
      J = 1
      IF (DECPT) ITYPE = 5
 1130 IF (IMAGE(I) .EQ. IEQUAL) GO TO 1200
 1140 IF (L .LT. NWORD) GO TO 1150
      IF (ITYPE .NE. 6) GO TO (1900, 1210), J
      IF (BCDFLG) I = I - 1
      GO TO (1500, 1210), J
 1150 FLD(L,NCHAR,NAME) = FLD(0,NCHAR,IMAGE(I))
      L = L + NCHAR
      GO TO (1900, 1140), J
 1160 IF (BCDFLG) GO TO 1170
      IF (IMAGE(I) .EQ. ICOMMA) GO TO 1200
      GO TO 1900
 1170 IF (IMAGE(I) .NE. IQUOTE) GO TO 1140
      NQT = 1 - NQT
      IF (NQT .EQ. 0) GO TO 1140
      IF (IMAGE(I+1) .NE. IQUOTE) BCDFLG = .FALSE.
      GO TO 1900
 1200 J = 2
      K = I
      I = 81
      GO TO 1140
 1210 I = K
      IF (ITYPE .GE. 5) GO TO 1500
      NNAMES = NAMTAB(1)
      INPFLG = 0
      NAMSAV = NAME
      DO 1220 K = 1, NNAMES
      IF (NAME .NE. NAMTAB(K+1)) GO TO 1220
      IF (ITYPE .EQ. 0) GO TO 1260
      J = 1
      GO TO 1999
```

OD 52258

SUBROUTINE INPUT(NAMTAB,DATA,INPFLG,OUTFLG)

```
 1220 CONTINUE                                                          INPUT 71
      GO TO 1996                                                        INPUT 72
 1260 NL = NNAMES + K + 1                                               INPUT 73
      NM = NNAMES + NL                                                  INPUT 74
      NC = NNAMES + NM                                                  INPUT 75
      NAMTAB(NC) = 0                                                    INPUT 76
      IF (IMAGE(1) .EQ. IEQUAL) GO TO 1600                              INPUT 77
      J = 2                                                             INPUT 78
      NAMTAB(NC) = NTRUE                                                INPUT 79
      GO TO 1998                                                        INPUT 80
 1300 IF (K .GT. 10) GO TO 1320                                         INPUT 81
      IF (.NOT. EXPNT) GO TO 1310                                       INPUT 82
      IEXP = IEXP * 10 + SIGN(K - 1, IESIGN)                            INPUT 83
      GO TO 1900                                                        INPUT 84
 1310 IF (DECPT) L = L + 1                                              INPUT 85
      VALUE = VALUE * D10 + DBLE(FLOAT(SIGN(K - 1, IVSIGN)))            INPUT 86
      GO TO 1900                                                        INPUT 87
 1320 K = K - 10                                                        INPUT 88
      GO TO (1330, 1330, 1350, 1360, 1380, 1900, 1390, 1325, 1335),K    INPUT 89
 1325 ITYPE = 3                                                         INPUT 90
 1330 IF (.NOT. DECPT) GO TO 1340                                       INPUT 91
      IF (K .EQ. 2) IESIGN = - 1                                        INPUT 92
 1335 EXPNT = .TRUE.                                                    INPUT 93
      GO TO 1900                                                        INPUT 94
 1340 IF (K .EQ. 2) IVSIGN = - 1                                        INPUT 95
      GO TO 1900                                                        INPUT 96
 1350 DECPT = .TRUE.                                                    INPUT 97
      ITYPE = 2                                                         INPUT 98
      GO TO 1900                                                        INPUT 99
 1360 IREPT = VALUE                                                     INPUT100
      GO TO 1620                                                        INPUT101
 1380 ITYPE = 4                                                         INPUT102
      GO TO 1900                                                        INPUT103
 1390 ITYPE = 6                                                         INPUT104
      BCDFLG = .TRUE.                                                   INPUT105
      J = 1                                                             INPUT106
      NOT = 0                                                           INPUT107
      GO TO 1900                                                        INPUT108
 1400 IF (INPFLG .NE. 0) GO TO 1600                                     INPUT109
      L = IEXP - L                                                      INPUT110
      IEXP = ABS(L)                                                     INPUT111
      DO 1430 K = 1, IEXP                                               INPUT112
      IF (L) 1410, 1440, 1420                                           INPUT113
 1410 VALUE = VALUE/D10                                                 INPUT114
      GO TO 1430                                                        INPUT115
 1420 VALUE = VALUE * D10                                               INPUT116
 1430 CONTINUE                                                          INPUT117
 1440 IF (ITYPE .NE. 4) GO TO 1500                                      INPUT118
      IF (CXBOTH) GO TO 1500                                            INPUT119
      CXVAL = VALUE                                                     INPUT120
      CXBOTH = .TRUE.                                                   INPUT121
      GO TO 1610                                                        INPUT122
 1500 IF (INPFLG .NE. 0) GO TO 1600                                     INPUT123
      DO  1590 K = 1, IREPT                                             INPUT124
 1505 INDEX = NAMTAB(NC) + 1                                            INPUT125
      IF (INDEX .GT. NAMTAB(NM)) GO TO 1997                             INPUT126
      NAMTAB(NC) = INDEX                                                INPUT127
      INDEX = INDEX + NAMTAB(NL)                                        INPUT128
      GO TO (1520, 1530, 1540, 1540, 1560, 1580, 1570), ITYPE           INPUT129
 1520 IVALUE = VALUE                                                    INPUT130
 1525 DATA(INDEX) = AVALUE(1)                                           INPUT131
      GO TO 1590                                                        INPUT132
 1530 DATA(INDEX) = VALUE                                               INPUT133
      GO TO 1590                                                        INPUT134
 1540 INDEX = INDEX + NAMTAB(NC) - 1                                    INPUT135
      IF (ITYPE .EQ. 4) GO TO 1550                                      INPUT136
      DATA(INDEX+1) = AVALUE(2)                                         INPUT137
      GO TO 1525                                                        INPUT138
 1550 DATA(INDEX) = CXVAL                                               INPUT139
      DATA(INDEX+1) = VALUE                                             INPUT140
```

APPENDIX G

SUBROUTINE INPUT(NAMTAB,DATA,INPFLG,OUTFLG)

```
      GO TO 1590                                               INPUT141
1560  LVALUE = .FALSE.                                         INPUT142
      IF (NAME .EQ. ITRUE) LVALUE = .TRUE.                     INPUT143
      GO TO 1525                                               INPUT144
1570  M = INDEX - IEXP                                         INPUT145
      DATA(INDEX) = DATA(M)                                    INPUT146
      IF (M .EQ. L) GO TO 1585                                 INPUT147
      GO TO 1505                                               INPUT148
1580  DATA(INDEX) = AVALUE(1)                                  INPUT149
      IEXP = IEXP + 1                                          INPUT150
      IF (BCDFLG) GO TO 1610                                   INPUT151
      ITYPE = 7                                                INPUT152
1585  L = INDEX                                                INPUT153
1590  CONTINUE                                                 INPUT154
1600  CXBOTH = .FALSE.                                         INPUT155
      IREPT = 1                                                INPUT156
      J = 2                                                    INPUT157
      K = 0                                                    INPUT158
      ITYPE = 1                                                INPUT159
1610  L = 0                                                    INPUT160
      IEXP = 0                                                 INPUT161
      IESIGN = 0                                               INPUT162
      IVSIGN = 0                                               INPUT163
      DECPT = .FALSE.                                          INPUT164
      EXPNT = .FALSE.                                          INPUT165
1620  VALUE  =  0.                                             INPUT166
1900  IF (I .LT. MAXCOL) GO TO 1100                            INPUT167
      IF (K .NE. 0) GO TO 1105                                 INPUT168
      GO TO 1030                                               INPUT169
1995  IF (INPFLG .NE. 0) GO TO 1900                            INPUT170
      INPFLG = 1                                               INPUT171
1996  INPFLG = 1 + INPFLG                                      INPUT172
1997  INPFLG = 1 + INPFLG                                      INPUT173
      J = 4                                                    INPUT174
1998  ITYPE = 0                                                INPUT175
1999  RETURN                                                   INPUT176
2000  IF (INPFLG .EQ. 0) GO TO 1600                            INPUT177
      IF (.NOT. OUTFLG) PRINT 2001, IMAGE                      INPUT178
2001  FORMAT (20X,80A1)                                        INPUT179
      DO 2010 J = 1, 80                                        INPUT180
      IMAGE(J) = IBLANK                                        INPUT181
2010  CONTINUE                                                 INPUT182
      IMAGE(IIMAGE) = NUMBER(14)                               INPUT183
      PRINT 2001, IMAGE                                        INPUT184
      J = 2                                                    INPUT185
      GO TO (2020, 2030, 2040), INPFLG                         INPUT186
2020  PRINT 2021, NAMSAV                                       INPUT187
2021  FORMAT (20X, 17HTOO MUCH DATA IN , A6/)                  INPUT188
      GO TO 1999                                               INPUT189
2030  PRINT 2031, NAMSAV                                       INPUT190
2031  FORMAT (20X, A6, 17H NOT IN NAME LIST/)                  INPUT191
      GO TO 1999                                               INPUT192
2040  PRINT 2041                                               INPUT193
2041  FORMAT (20X, 33HSYNTAX ERROR OR ILLEGAL CHARACTER/)      INPUT194
      GO TO 1999                                               INPUT195
      END                                                      INPUT196
```

I N D E X   PAGE 56

SUBROUTINE INPUT(NAMTAB,DATA,INPFLG,OUTFLG)

SYMBOL		REFERENCES									
1030	-	23	24*	169							
1100	-	28*	167								
1105	-	32*	168								
1110	-	31	33*								
1120	-	33	34	37*							
1130	-	31	41*								
1140	-	42*	48	52	54	60					
1150	-	42	46*								
1160	-	29	49*								
1170	-	49	52*								
1200	-	32	41	50	57*						
1210	-	43	45	61*							
1220	-	66	67	71*							
1260	-	23	68	73*							
1300	-	35	36	81*							
1310	-	82	85*								
1320	-	81	88*								
1325	-	89	90*								
1330	-	89	91*								
1335	-	89	93*								
1340	-	91	95*								
1350	-	89	97*								
1360	-	89	100*								
1380	-	89	102*								
1390	-	89	104*								
1400	-	32	109*								
1410	-	113	114*								
1420	-	113	116*								
1430	-	112	115	117*							
1440	-	113	118*								
1500	-	45	62	118	119	123*					
1505	-	125*	146								
1520	-	129	130*								
1525	-	131*	138	144							
1530	-	129	133*								
1540	-	129	135*								
1550	-	136	139*								
1560	-	129	142*								
1570	-	129	145*								
1580	-	129	149*								
1585	-	147	153*								
1590	-	124	132	134	141	154*					
1600	-	23	27	77	109	123	155*	177			
1610	-	122	151	160*							
1620	-	101	166*								
1900	-	30	43	48	51	56	84	87	89	94	96
		99	103	108	167*	170					
1995	-	38	170*								
1996	-	72	172*								
1997	-	126	173*								
1998	-	80	175*								
1999	-	70	176*	189	192	195					
2000	-	23	177*								
2001	-	25PR	178PR	179*	184PR						
2010	-	180	182*								
2020	-	186	187*								
2021	-	187PR	188*								
2030	-	186	190*								
2031	-	190PR	191*								
2040	-	186	193*								
2041	-	193PR	194*								
ABS	-	111									
AVALUE	-	10DI	11EQ	131	137	149					
BCDFLG	-	8LG	44	49	55*	105*	151				
CINCOD	-	3CL									
CMACHN	-	6CL									
CXBOTH	-	8LG	119	121*	155*						
CXVAL	-	120*	139								

APPENDIX G

```
I N D E X                                                               PAGE  57
                         SUBROUTINE INPUT(NAMTAB,DATA,INPFLG,OUTFLG)

D10      -     9DB     13DA     86      114      116
DATA     -     1AG     10DI     131=    133=     137=     139=     140=     146=     149=
DPLE     -    86
DECPT    -     8LG     36       40      85       91       97=      164=
EXPNT    -     8LG     82       93=     165=
FLD      -    46=
FLOAT    -    86
FRMAT1   -    13DA     24RD
I        -    11EQ     26=      28=     30       31       34       41       44=      50       52
              55       58       59=     61=      77       167
IBLANK   -     3CO     30       181
ICOMMA   -     4EQ     31       50
IEQUAL   -     5EQ     41       77
IESIGN   -    83       92=      162=
IEXP     -    83=      110      111=    112      145      150=     161=
IIMAGE   -     3CO     11EQ     183
IMAGE    -     3CO     24RD     25PR    30       31       34       41       50       52       55
              77       178PR    181=    183=     184PR
INAM     -     3CO
INDEX    -    125=     126      127     128=     131      133      135=     137      139      140
              145      146      149     153
INPFLG   -     1AG     64=      109     123      170      171=     172=     173=     177      186
INPUT    -     1
IQUOTE   -     4EQ     52       55
IREPT    -    100=     124      156=
ITRUE    -    13DA     143
ITYPE    -    13DA     29       40=     43       62       68       90=      98=      102=     104=
              118      129      136     152=     159=     175=
IVALUE   -    11EQ     130=
IVSIGN   -    86       95=      163=
J        -    13DA     23       31      32       39=      43       45       48       57=      69=
              73=      106=     157=    174=     180=     181      185=
K        -    33=      34       35      58=      61       66=      67       73       81       83
              86       88=      89      92       95       112=     124=     158=     168
L        -    38       42       46      47=      85=      110=     111      112      147      153=
              160=
LTRUE    -     8LG     12EQ     14DA
LVALUE   -     3LG     11EQ     142=    143=
M        -    145=     146      147
MAXCOL   -     6CO     167
NAME     -    11EQ     38       46      65       67       143
NAMSAV   -    65=      187PR    190PR
NAMTAB   -     1AG     10DI     63      67       76=      79=      125      126      127=     128
              135
NC       -    75=      76       79      125      127      135
NCHAR    -     6CO     46       47
NL       -    73=      74       128
NM       -    74=      75       126
NNAMES   -    63=      66       73      74       75
NQT      -    53=      54       107=
NTRUE    -    12EQ     79
NUMBER   -     3CO     4EQ      5EQ     34       183
NWORD    -     6CO     42
OUTFLG   -     1AG     8LG      25      178
RETURN   -    176
SIGN     -    93       86
VALUE    -     9DB     11EQ     86=     100      114=     116=     120      130      133      140
              166=
```

APPENDIX G

BLOCK DATA FOR INPUT SUBROUTINE

```
      BLOCK DATA                                                        BLOCK  1
      COMMON /CINCOD/ INAM,IIMAGE,IMAGE(80),IBLANK,NUMBER(21)           BLOCK  2
      DATA IBLANK/1H /                                                  BLOCK  3
      DATA (NUMBER(I),I=1,21)/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,  BLOCK  4
     1                 1H+,1H-,1H.,1H*,1H(,1H),1H",1HD,1HE,1H,,1H=/     BLOCK  5
C                                                                       BLOCK  6
C        NOTE THAT THE FOLLOWING DATA IS MACHINE DEPENDENT.             BLOCK  7
C        NWORD = NO.OF BITS IN MACHINE WORD.                            BLOCK  8
C        NCHAR = NO.OF BITS IN ALPHA-NUMERIC CHARACTER.                 BLOCK  9
C        MAXCOL = NO.OF COLUMNS TO BE READ FROM CARDS.                  BLOCK 10
C                                                                       BLOCK 11
      COMMON /CMACHN/ NWORD,NCHAR,MAXCOL                                BLOCK 12
      DATA NWORD/36/,NCHAR/6/,MAXCOL/80/                                BLOCK 13
      END                                                               BLOCK 14
```

APPENDIX G

```
I N D E X                                                                  PAGE  59
                              BLOCK DATA FOR INPUT SUBROUTINE

SYMBOL      * * * * * * * * * * * * * * *   REFERENCES   * * * * * * * * * * * * * * *

CINCOD  -    2CL
CPACNN  -   12CL
I       -    4DA
IBLANK  -    2CO     30A
IIMAGE  -    2CO
IPAGE   -    2CO
INAM    -    2CO
MAXCOL  -   12CO    13DA
NCHAR   -   12CO    13DA
NUMBER  -    2CO     4DA
NWORD   -   12CO    13DA
1BLCKO  -    1
```

APPENDIX G

OD 52258

********** SUPER INDEX **********

SYMBOL - ****************** ROUTINES IN WHICH THE SYMBOL IS USED ******************

A	-	FNCNS	RBCOMP	RBSORT				
A1	-	RVCOMP						
A2	-	RVCOMP						
ABS	-	BCOMP	FNCNS	INPUT	RBCOMP	RBSORT	RVCOMP	RVSPRD
ACOS	-	FNCNS						
AINT	-	BCOMP						
AINTRP	-	RBCOMP	TABLKP					
ALOG10	-	RTCOMP						
ALPHC	-	BCOMN1	RVCOMP					
AMAX1	-	RBCOMP	RVCOMP					
AMIN1	-	RBCOMP						
AMOD	-	RBCOMP	RVCOMP					
AR1	-	BCOMN1	IDENT	RBSORT	1BLCK	1MAIN		
ARG	-	FNCNS	RBCOMP	TABLKP				
ARRAY	-	SORT						
ATAN2	-	RBCOMP						
AVALUE	-	INPUT						
B	-	BCOMP	RVCOMP					
BAND	-	BCOMP	BCOMN1	RBCOMP	RVCOMP			
BCDFLG	-	INPUT						
BCOMP	-	1MAIN						
BND	-	BCOMP						
BNDOUT	-	BCOMP	BCOMN1	RVPRNT				
BR1	-	BCOMN1	IDENT	RBCOMP	RBSORT	TCOMP	1BLCK	1MAIN
BWIDTH	-	BCOMN1	IDENT	RVPRNT				
BWINT	-	BCOMP	BCOMN1	IDENT				
CD	-	BCOMN1	IDENT	RVCOMP	RVPRNT			
CORT	-	BCOMP	BCOMN1	IDENT	RBCOMP			
CALFAP	-	RBCOMP						
CALFBP	-	RBCOMP						
CBAND	-	BCOMN1						
CCOUNT	-	BCOMN1						
CENTER	-	BCOMP	BCOMN1					
CFCNST	-	BCOMN1						
CGAMAP	-	RBCOMP						
CGAMBP	-	RBCOMP						
CHCNST	-	BCOMN1						
CINCOD	-	INPUT	1BLCKD					
CINDAT	-	BCOMN1						
CINDEF	-	BCOMN1						
CINPUT	-	BCOMN1	FNCNS					
CMACHN	-	INPUT	1BLCKD					
CNT	-	BCOMP						
CONTMU	-	RBCOMP						
COS	-	BCOMP	FNCNS	IDENT	RBCOMP	RVCOMP		
COSA	-	FNCNS	RBCOMP					
COSB	-	FNCNS	RBCOMP					
COSC	-	FNCNS						
COSORT	-	BCOMN5	RBCOMP					
COSTHA	-	BCOMN5	RBCOMP					
COSTHB	-	BCOMN5	RBCOMP					
CPRINT	-	BCOMN1						
CSA	-	RVCOMP						
CSALFA	-	RBCOMP						
CSALFB	-	RBCOMP						
CSBETA	-	RBCOMP						
CSBETB	-	RBCOMP						
CSGAMA	-	RBCOMP						
CSGAMB	-	RBCOMP						
CSKSI	-	BCOMN1	IDENT	RBCOMP	RVCOMP			
CSOPTD	-	RBCOMP						
CSOMTF	-	RVCOMP						
CSPHI	-	RVCOMP						
CSPRED	-	BCOMN1						
CSTHAD	-	RBCOMP						
CSTHBD	-	RBCOMP						
CTAPE	-	BCOMN1						
CTBLKP	-	AINTRP	BCOMN1	TABLKP				

APPENDIX G

********** SUPER INDEX **********

CTHTMP	-	RBCOMP							
CXBOTH	-	INPUT							
CXCNST	-	DCOMN1							
CXVAL	-	INPUT							
D	-	BCOMP	RESORT	RVCOMP					
DC	-	DCOMN1	RVPRNT						
D1C	-	INPUT							
DATA	-	INPUT							
DBLE	-	INPUT							
DBTTP	-	DCOMN1	RESORT	RVPRNT					
DCOMN1	-	BCOMP	IDENT	RBCOMP	RBSORT	RTCOMP	RVCOMP	RVPRNT	RVSPRD
		TCOMP	1BLCK	1MAIN					
DCOMN2	-	RBCOMP	RTCOMP	RVCOMP	RVPRNT	RVSPRD	1MAIN		
DCOMN3	-	RBSORT	TCOMP						
DCOMN4	-	RVCOMP							
DCOMN5	-	RBCOMP							
DCOMN6	-	1BLCK							
DD	-	RBCOMP	RVCOMP						
DECPT	-	INPUT							
DEGRAD	-	BCOMP	DCOMN1	IDENT	RBCOMP	1BLCK			
DELDEP	-	AINTRP	DCOMN1						
DELIND	-	DCOMN1	TABLKP						
DELR	-	RBCOMP							
DELT	-	DCOMN1	IDENT	RBCOMP	RVCOMP	TCOMP			
DELTZ	-	DCOMN1	IDENT	RVPRNT					
DEPND	-	AINTRP	TABLKP						
DOP	-	DCOMN5	RBCOMP						
DOPSKP	-	RBCOMP							
DR	-	DCOMN1	IDENT	RVCOMP					
E	-	BCOMP							
END	-	DCOMN1	IDENT	1MAIN					
EXP	-	IDENT	RBCOMP	RVCOMP					
EXPNT	-	INPUT							
EXPS	-	DCOMN1	IDENT	RVCOMP					
F	-	BCOMP							
FC	-	DCOMN1	RBCOMP	RVCOMP					
F1	-	BCOMP	DCOMN1	IDENT	RBCOMP	RTCOMP	RVCOMP	1BLCK	
F1C	-	DCOMN1	IDENT	RTCOMP	RVCOMP	1BLCK			
F18D	-	BCOMP	DCOMN1	IDENT	1BLCK				
F1E3	-	DCOMN1	IDENT	1BLCK					
F1MIN	-	DCOMN1	1BLCK						
F2	-	DCOMN1	IDENT	RBCOMP	1BLCK				
F2C	-	DCOMN1	RBCOMP	1BLCK					
F3	-	DCOMN1	1BLCK						
F4	-	DCOMN1	IDENT	1BLCK					
F9C	-	BCOMP	DCOMN1	IDENT	RBCOMP	1BLCK			
FA	-	RBCOMP							
FACTOR	-	AINTRP	DCOMN1	TABLKP					
FCC3	-	DCOMN1	IDENT	RVCOMP					
FCCVS	-	DCOMN1	IDENT	RBCOMP	RVCOMP				
FCNS	-	FNCNS							
FCSGAM	-	DCOMN4	RVCOMP						
FGAM	-	DCOMN1	RVCOMP						
FILTER	-	BCOMP	DCOMN1	IDENT	RTCOMP				
FLD	-	INPUT							
FLOAT	-	INPUT							
FL0610	-	DCOMN1	IDENT	RBCOMP	RVCOMP	1BLCK			
FNBAND	-	BCOMP	DCOMN1	RVCOMP					
FPTS	-	BCOMP	DCOMN1	RBCOMP	RVCOMP	TCOMP	1BLCK		
FR	-	RVCOMP							
FREQ	-	FNCNS							
FRMAT1	-	INPUT							
FRSQ	-	RVCOMP							
FRV	-	RVCOMP							
FX	-	RVCOMP							
FXSQ	-	RVCOMP							
FZ2	-	DCOMN1	IDENT	RVCOMP					
FZRO	-	BCOMP	DCOMN1	IDENT	RBCOMP	RVCOMP	RVPRNT		
FZSQ	-	DCOMN1	IDENT	RVCOMP					
G	-	RVCOMP							

APPENDIX G

OD 52258


```
                              ********** SUPER INDEX **********

GO       -  DCOMN1   IDENT
HBAND    -  DCOMN1   IDENT    RVPRNT
HEADS    -  DCOMN1   IDENT    1BLCK
HED      -  DCOMN1   IDENT    RVPRNT   1BLCK
HFROM    -  DCOMN1   IDENT
HOUTPT   -  DCOMN1   RVPRNT
HRVB     -  RVPRNT
HSPRD    -  DCOMN1   IDENT
HSPRW    -  DCOMN1   IDENT
HTIME    -  RVPRNT
HTOT1    -  DCOMN1   RVPRNT   1BLCK
HTOT2    -  DCOMN1   RVPRNT   1BLCK
HTOT3    -  DCOMN1   IDENT    RVPRNT   1BLCK
HUNIT    -  DCOMN1
I        -  BCOMP    IDENT    INPUT    RBCOMP   RBSORT   RTCOMP   RVCOMP   RVPRNT
            RVSPRD   SORT     TCOMP    1BLCK    1BLCKO
IBLANK   -  DCOMN1   INPUT    1BLCKO
IBOUND   -  TCOMP
ICOMMA   -  INPUT
ID       -  IDENT
IDATA    -  DCOMN1   IDENT
IDATE    -  DCOMN1   RVPRNT
IDC      -  DCOMN1   IDENT    RVPRNT
IDENT    -  1MAIN
IDOP     -  RBCOMP
IDV      -  DCOMN1   RVPRNT
IEQUAL   -  INPUT
IESIGN   -  INPUT
IEXP     -  INPUT
IFIX     -  BCOMP    RVCOMP
IFLAG    -  FNCNS
IFZRO    -  DCOMN1
II       -  RBCOMP   RVPRNT   RVSPRD
IIMAGE   -  INPUT
IJ       -  RBCOMP
IMAGE    -  INPUT
IMAX     -  RTCOMP
INDEP    -  TABLKP
INDEX    -  INPUT
INFNT    -  DCOMN1   1BLCK
INPFLG   -  IDENT    INPUT
INPT     -  DCOMN1   1BLCK
INPUT    -  IDENT
IPEVRY   -  BCOMP    DCOMN1   RVPRNT   TCOMP    1BLCK
IPLT     -  DCOMN1   RVPRNT   1BLCK    1MAIN
IPRT     -  DCOMN1   IDENT    RVPRNT   1BLCK
IQUOTE   -  INPUT
IREPT    -  INPUT
ISIGN    -  RBCOMP
ITABLE   -  AINTRP   DCOMN1   RBCOMP   TABLKP   TCOMP
ITABMN   -  TABLKP   1BLCK
ITABMX   -  TABLKP
ITOTAL   -  RVPRNT
ITRUE    -  INPUT
ITYPE    -  INPUT
IVALUE   -  INPUT
IVSIGN   -  INPUT
J        -  IDENT    INPUT    RBSORT   RTCOMP   RVCOMP   RVPRNT   RVSPRD   SORT
            TCOMP    1BLCK
JA       -  RBCOMP
JB       -  RBCOMP
JBOUND   -  RBCOMP
JC       -  RBCOMP
JD       -  RBCOMP
JDD      -  RBCOMP
JDELT    -  RBCOMP
JDMAX    -  RBCOMP
JDMIN    -  RBCOMP
JE       -  RBCOMP
JMAX     -  RBCOMP
```

********** SUPER INDEX **********

JUMP	-	RBCOMP							
K	-	IDENT	INPUT	RBSORT	RVCOMP	RVPRNT	RVSPRD	TCOMP	
K0	-	DCOMN1	1BLCK						
K1	-	BCOMP	DCOMN1	IDENT	RBCOMP	RTCOMP	RVCOMP	RVPRNT	RVSPRD
		TCOMP	1BLCK	1MAIN					
K10	-	DCOMN1	IDENT	1BLCK					
K2	-	DCOMN1	IDENT	RBCOMP	RTCOMP	RVCOMP	TCOMP	1BLCK	
K3	-	DCOMN1	IDENT	RBCOMP	RTCOMP	RVSPRD	TCOMP	1BLCK	
K40	-	DCOMN1	RVPRNT	TCOMP	1BLCK				
K5	-	DCOMN1	IDENT	RVPRNT	1BLCK				
K6	-	DCOMN1	1BLCK						
K8	-	DCOMN1	RVPRNT	1BLCK					
KBAND	-	RBCOMP							
KK	-	RVSPRD							
KSI	-	DCOMN1	IDENT						
KSID	-	DCOMN1	IDENT	RVPRNT					
KT	-	DCOMN1	RBCOMP	RTCOMP	RVCOMP	1MAIN			
KTIM	-	RBCOMP							
KTSBND	-	BCOMP	DCOMN1	IDENT					
KTT	-	DCOMN1	RBCOMP	RTCOMP	WVCOMP	1MAIN			
L	-	BCOMP	IDENT	INPUT	RBCOMP	RBSORT	RVPRNT		
LEAP	-	RBCOMP							
LENGTH	-	SORT							
LFLAGS	-	DCOMN1	1BLCK						
LL	-	RBCOMP							
LMBAND	-	BCOMP	DCOMN1	DCOMN6	RVSPRD				
LMBND1	-	DCOMN1	DCOMN6	RTCOMP	RVPRNT				
LMKS	-	DCOMN1	DCOMN6	RBSORT					
LMKS2	-	DCOMN1	DCOMN6	RBCOMP					
LMNT	-	DCOMN1	1MAIN						
LMRB	-	DCOMN1	DCOMN6	1MAIN					
LMSPRD	-	DCOMN1	DCOMN6						
LMTIM	-	DCOMN1	DCOMN6	TCOMP					
LMTRS	-	DCOMN1	DCOMN6	TCOMP					
LOG4PI	-	DCOMN1	IDENT	1BLCK					
LOGMV	-	DCOMN1	IDENT						
LOGMVI	-	DCOMN1	IDENT	RVCOMP					
LTRUE	-	INPUT							
LVALUE	-	INPUT							
M	-	IDENT	INPUT	RBSORT	TCOMP				
MAXO	-	IDENT							
MAXCOL	-	INPUT	1BLCKO						
MBAND	-	BCOMP	DCOMN1						
MINO	-	BCOMP	RVCOMP	1MAIN					
ML	-	RBCOMP							
MLMU	-	RBCOMP							
MM	-	RBCOMP							
MN	-	RBCOMP							
MNBND	-	BCOMP	DCOMN1	RTCOMP	RVPRNT	TCOMP			
MNMX	-	RBCOMP							
MOD	-	BCOMP	RVPRNT						
MU	-	RBCOMP							
MX	-	RBCOMP							
N	-	RBSORT							
NA	-	RBCOMP	RBSORT						
NAMCNT	-	DCOMN1	IDENT	1BLCK					
NAMDAT	-	DCOMN1	1BLCK						
NAME	-	INPUT							
NAMSAV	-	INPUT							
NAMTAB	-	INPUT							
NB	-	RBCOMP	RBSORT						
NBAND	-	BCOMP	DCOMN1	RBCOMP	RVCOMP	RVSPRD			
NBEAM	-	DCOMN1	FNCNS						
NBOUND	-	DCOMN1	RBCOMP	TCOMP					
NBSPRD	-	DCOMN1	IDENT						
NBTM	-	DCOMN1	RBSORT						
NC	-	INPUT	RBSORT						
NCHAR	-	INPUT	1BLCKO						
ND	-	RBSORT							
NEXT	-	RBCOMP							

********** SUPER INDEX **********

APPENDIX G

********** SUPER INDEX **********

RBXB	-	DCOMN5	RBCOMP						
RD	-	RBCOMP							
RECV	-	DCOMN1	RBCOMP	RVCOMP					
RELBND	-	BCOMP	DCOMN1						
RETURN	-	AINTRP	BCOMP	FNCNS	IDENT	INPUT	RBCOMP	RBSORT	RTCOMP
		RVCOMP	RVPRNT	RVSPRD	SORT	TABLKP	TCOMP		
REVERB	-	DCOMN2	1MAIN						
RRF	-	BCOMP	FNCNS						
RRFS	-	BCOMP	DCOMN1	RTCOMP					
RT	-	RBCOMP							
RTCOMP	-	1MAIN							
RV	-	DCOMN2	RBCOMP	RVSPRD					
RVB	-	DCOMN2	RTCOMP	RVPRNT					
RVCOMP	-	1MAIN							
RVPRNT	-	1MAIN							
RVS	-	DCOMN2	RTCOMP	RVPRNT					
RVSPRD	-	1MAIN							
RVT	-	DCOMN2	RTCOMP	RVPRNT					
RVV	-	DCOMN2	RTCOMP	RVCOMP	RVPRNT				
S	-	DCOMN1	IDENT	RBSORT	RVPRNT				
SHIFT	-	DCOMN1	RBSORT	1BLCK					
SIGN	-	BCOMP	INPUT	RBCOMP	RVCOMP				
SIN	-	IDENT	RBCOMP	RVCOMP					
SINOMT	-	DCOMN5	RBCOMP						
SNA	-	RVCOMP							
SNKSI	-	DCOMN1	IDENT	RBCOMP	RVCOMP				
SNOMTD	-	RBCOMP							
SNOMTF	-	RVCOMP							
SNPHI	-	RVCOMP							
SONTMU	-	RBCOMP							
SORT	-	RBCOMP	TCOMP						
SPRCMP	-	IDENT							
SPREAD	-	DCOMN1	IDENT	RTCOMP	RVPRNT	TCOMP	1MAIN		
SPRED	-	DCOMN1	RVSPRD						
SQRT	-	RBCOMP	RVCOMP						
STOP	-	1MAIN							
T	-	DCOMN5	RBCOMP	RTCOMP					
T1	-	RVCOMP							
T2	-	RVCOMP							
T3	-	RVCOMP							
TABLKP	-	RBCOMP	TCOMP						
TCOMP	-	1MAIN							
TD	-	RBCOMP							
TEMP	-	SORT							
TEST	-	RBSORT							
TMA	-	DCOMN5	RBCOMP						
TMAD	-	RBCOMP							
TMAT	-	RBCOMP							
TMB	-	DCOMN5	RBCOMP						
TMBD	-	RBCOMP							
TMBT	-	RBCOMP							
TMTMAX	-	BCOMP	DCOMN1	IDENT	RBCOMP				
TIMCMP	-	DCOMN1	RBSORT	TCOMP					
TIME	-	DCOMN1	FNCNS	RBCOMP	RTCOMP	RVCOMP	RVPRNT	TCOMP	
TRIN	-	DCOMN3	RBSORT	TCOMP					
TOTALS	-	DCOMN1	IDENT	RVPRNT	TCOMP				
TR	-	RTCOMP							
TRS	-	DCOMN1	TCOMP	1BLCK					
TRS2	-	DCOMN1	1BLCK						
TRS3	-	DCOMN1	1BLCK						
TTMAX	-	RBCOMP							
TVG	-	DCOMN1	IDENT	RTCOMP					
TVGF	-	FNCNS	RTCOMP						
TWMAX	-	RBCOMP							
TWMIN	-	RBCOMP							
TWOPI	-	DCOMN1	IDENT	RBCOMP	RVCOMP	1BLCK			
VALUE	-	INPUT							
VOL1	-	RVCOMP							
VPTTRN	-	BCOMP	DCOMN1	RVCOMP					
VS	-	BCOMP	DCOMN1	IDENT	RBCOMP	RVPRNT			

OD 52258

********** SUPER INDEX **********

****

I N D E X
END OF COMPUTATION,
6 APRIL 1976 VERSION.

(AS MODIFIED AT NUC BY P. MARSH, M. ALLIES, M. PINNEY, ET AL)

APPENDIX G